UPDATE! $4.95 OCTOBER 1989

ComputerSystems

Take it to the Limit

A Quarterly Magazine for the users

EXPLORING THE POWER

OF

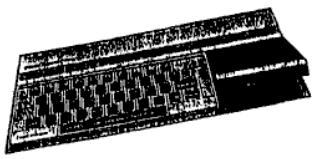
SINCLAIR COMPUTERS

THE CAMBRIDGE Z88
Portable Power

THE SINCLAIR QL
32 Bit Power House

THE SINCLAIR TIMEX TS-2068
Versatile Desk Topper

In this issue. . .



Automatic Printer Control Utilities

The June 1990 Milwaukee Sinclair Fest

PLUS, a magazine full of Programs and utilities

UPDATE MAGAZINE
1317 Stratford Ave.
Panama City, FL 32404
904 871 3556

NEWS AND NEW PRODUCTS

QL DISK SYSTEMS

3.5" Drive Systems

Dual 3.5" Mitsubishi 400K-

This great little system includes 2-3.5" Mitsubishi SSQD drives, case w/ps, cable w/ends and interface. Complete package!

READY TO PLUG AND GO!

ORDER RMG#P103 $325+$8 PH

Dual 3.5" Teac 800K System-

Includes 2 Teac FD135 DSQD drives, case w/ps, cable & interface. HUGE CAPACITY!

ORDER RMG#P119 $425+$8 PH

Many more packages available!

Just call or write with what you want. Prices start at only $199.95 plus PH. Individual components also available.

5.25" Drive System

Dual 5.25" 800K System-

This system comes complete with the disk interface, dual 5.25" DSQD drives, cable and case with power supply. This gives you a mass storage area of 1,440,000 bytes of data and programming. We set up and test EVERY drive set that we sell!

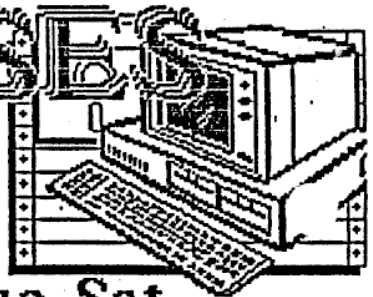
Alternate Interfaces-

If you want expanded memory with your QL disk system, we can supply TRUMP CARDS starting as low as $250 (256K). Other sizes available upon request.

WRITE FOR INFO OR CATALOG ($3)

For ALL of your computer needs, CALL-

RMG ENTERPRISES



1419 1/2 7th Street
Oregon City, OR 97045
503/655-7484 Noon-10PM Pacific Tue-Sat

Editorials and Announcements

Update QL Editor

Update is please to announce that Bill Cable is now the Sinclair QL EDITOR for Update Magazine. As most subscribers know, Bill Cable is a foremost QL Programmer and has contributed emensely to the content of the magazine. Now Bill will be contributing even more. Most importantly, we now have someone who will maintain accuracy in the articles and programs for the QL. Your scribe has been uneasy about that, given his novice status with the QL. Welcome Bill Cable, and thanks for your dedication to the QL Users of North America.

TS-2068 and Z88 editing will continue to be handled here on board the Update Yacht. Once in a while we may be a couple of weeks late, as the vessel gets laid up by hurricanes or the doldrums. Another writer for the Z88 has come aboard. Mike Fink (Domino Cubes) is a Z88 dealer that has gotten rich selling Z88's and periphreals. Now Mike will be doing a regular column in Update keeping us abreast of the happenings with this nice portable. If you miss Syd Wyncoop in this issue, not to worry. Syd DID send in an article but Update was already full for October. We'll enjoy that one in the next issue.

One subscriber wrote saying that he skims through Update for "just the articles about his computer and its disk system". Well heck! He is missing out on the educational benefits AND the possibility of actually using the programs given for the "foreign computer". Most programs for the QL, TS-2068 and Z88 can be kludged to work with the other. BASIC is BASIC. QL Super Basic, BBC Basic, and TS-2068 Basic are but variations of the same language, kinda like my southern drawl and Pete Hale's Bostonian twang. This issue has an article by Earl Dunnington that illustrates the point. Earl took our "LKDOS Tool by the Dozen" and now presents the same tools for TOS DISK. Another Update article in this issue "Printers and Printer Control" is generic in application and can be cloned in any computer's syntax and variation of Basic.

Actually both the QL and Z88 need such a printer control group utility to augument Quill and Pipedream. So, Update trys to give as many such programs and utilities to Public Domain as we can for the use of all. The TS-2068 Disk Drive users need not wait for a program that is specific for their DOS. All TS-2068 programs are adaptable to all DOS, with minor syntax changes.

Expanding Cooperation

Update has quietly dropped Commerical Advertising in favor of a concept of Mutual Cooperation. I looked back two years and found that the income from advertising has been nil. Then I got to thinking. Most Dealers are operating as a Hobby, or at such low key that cannot support much advertising. Programmers can spend as much on advertising as they realize from the sales that the ads bring in. Subscribers do need to be exposed to SOURCE information. "Where can I find a whatsit for my computer"? We ALL need continuing support. So Update's diabolical new scheme is to provide FREE advertising as a cooperative public service. The way it is put to the dealers and programmers is: "Think of something that you can do in return for your free ad".

Examples: Rod Gowen (RMG Enterprises) includes an Update Flier in his catalog mail outs. Ed Grey (Ed Grey Enterprises) posts a notice for Update in his BBS. Peter Hale (EMSOFT) and Bill Cable (Wood And Wind Computers) were the first to join into the scheme. Mark Stuber (Sharps Inc.) has been contacted to cooperate in similiar fashion. Individual programmers may not have much to offer in return, except their good will, which is sufficient. The main idea is to provide the most support possible in a cooperative effort.

Publication Situation

THE QUITS: Now that other publications have pulled the plug I get quite a lot of questions about the future of Update Magazine. It is almost as if folks are thinking "Update is next". Well, all that I can say is that Update is committed for another year of publishing- BUT. There is always that qualifier, eh? The publishing of Update is a hobby for me, and hobbies can become entanglements that interfere with other life needs, such as vacations, etc. AND, everyone is subject to the burn outs. I will attempt to keep Update going indefinitely by throwing out this gauntlet. If YOU are dedicated to Sinclair User Support, and have the ability to publish, then volenteer to take over publishing Update at some time in the future. That wont be like Ford taking over General motors, with accompanying stock market rumbles. It will be an easy acquisition and a smooth resumption. Call me.

The Cambridge Z88

Whither goest?

I will try to make this a thoughtful and honest appraisal of the Cambridge 88 Computer, and how I see its future developing. First, to compare the Z88 with other portable computers in its price range, and we wont go into a lot of figures to do that. The Z88 compares almost equally to the Tandy Portable in both price and capability. Both are "8 bit CPU" computers, which means that they are capable of managing only 64K of Memory at one time. This also means that neither computer is capable of being "Compatable With the Operating system of IBM Computers which is "PC Basic and MS-DOS". The operating system for both the IBM and the Apple Macintosh are for 16 and 32 bit CPU operation. So we must first explode this myth that seems to be developing. The Z88 IS NOT IBM or MACINTOSH COMPATABLE. NONE OF THE SOFTWARE FOR IBM or MACINTOSH COMPUTERS WILL WORK WITH THE Z88 or the TANDY PORTABLE.

Speaking of CAPABILITY, and comparing Capability with other computers, the Z88 compares with the TS-2068, the Sinclair Spectrum, the Commodore 128, and other 8 bit computer systems. All comparisons with those computers will result in the Z88 showing superiority in all areas except Screen Dislay and Interface to outside devices. The latter is a serious handicap with the Z88, but there has been a recently developed Portable Disk drive which mitigates that problem. The screen display is a problem yet to be resolved with both the Z88 and its counterpart, the Tandy Portable.

Now lets turn to the COST factor. Just how much the Z88 can be purchased for depends upon the deal that you can get from your dealer. My Z88, equipped with a 128K RAM cartridge, a 32K EPROM Cartridge, the PC LINK Eprom, and carrying case, cost around $700. A Commodore C-128, equipped with a disk drive and Monitor will cost slightly higher. I had a C-128 once and didn't like it at all. The Z88 has much more capability for managing extra Memory and has more overall computing capability. But the C-128 has the advantage of better display and a well established library of commercial software.

But we have been discussing and comparing a "desk top computer with a portable". Should we? Is there a difference in the Computing needs that one has if one needs a Portable? I think not. The Computer industry tends to agree. Portables are now being offered that are IBM MS-DOS compatable. So lets look at one line of portables. A current Computer Shopper ad shows a Toshiba T-1000 that can be a comparison. This one is an IBM Compatable that runs MS-DOS programs. Its fold-up LED display is larger than the Z88's display, and it has a built-in 720K Disk drive. The Toshiba 1000 also has 512K of RAM installed. The advertised price is $649.95. I wont comment upon how good or bad the Toshiba 1000 may or may not be. It is Tosyiba's "low end" portable. Toshiba's other portables run up to $5000 in price. There are other portables that are IBM compatables whose price range could form a basis for comparison with the Z88.

Enough Comparing! Lets see what the Z88's real attributes are. And, I might say that Cambridge North America is not pushing the real attributes of the Z88. Instead they seem to be hyping the ability to transfer ASCII Files back and forth between the Z88 and IBM computers. That Isn't "IBM COMPATABILITY". It is instead a "Software Innovation", that is commonly done every day with Telecom (Modem) software. The real attributes of the Z88 is in its operating system that can easily address up to One Megabyte of Extra Memory. This capability, which is un-matched by any existing 8 bit computer, has not been emphasised in the marketing efforts by Cambridge North America. They seem to be un-aware of the main attributes of the Z88 and are chasing a rainbow in the "IBM Compatability" forest.

Let us visualize that we all have IBM computers. You might be a travelling salesman and need a Portable Computer. Would you spend $6-800 to buy a Z88, knowing that it cannot run the library of software that you have for your IBM? Well, maybe the ability to transfer data files between the two computers might be worth the price? But no, you can get that low end Toshiba for about the same price, eh? And it can run your IBM software.

The Z88's competitive position seems to be right there with the "low end" computers like the Commodore 128 and Spectrum. But to be competitive there the Z88 needs better support by Sinclair Research and Cambridge North America. Needed are: Interfaces to Disk Drive, Interfaces to a desk top Monitor, and Software. While the two built-in softwares "Pipedream and Diary" are excellent, they hardly suffice as a complete computer system library. Such support would turn the Z88 from its

single purpose as a portable to both a portable and a desk top system, and could blow Commodore out of the low end computer market.

So, to evaluate the future of the Z88 in the North American market, we must speculate about the INTENT of Sinclair Research and their American representative, Cambridge North America.

Update Magazine began seeking cooperation from Cambridge North America before Christmas of 1988. A number of letters since have requested regular news releases of technical information, programming information, and general user support information. Copies of Update magazine have been sent to them. As of this time Cambridge North America has not cooperated at all. Though a small publication, Update Magazine is the only publication in North America that is willing to provide on-going Z88 User Support. This could help Cambridge North America in their marketing efforts by the building up of a data base about the Z88 and creating interest in the Z88. Satisfied and ehthusiastic users become the best advertising, and money cannot buy that.

That Cambridge North America would completely spurn and ignore our efforts, which would certainly be of benefit to them, is both enigmatic and ignominious. The only conclusion that can be drawn is that there is no interest on their part in USER SUPPORT, nor in building a reliable reputation in the North American market place. If Cambridge North America begins to show more interest in user support we will certainly credit them with a change in attitude and will welcome their participation. There is really no reason why they shouldn't. What is the cost of a few pieces of paper mailed out periodically to a magazine? If this article makes you a bit miffed, write them a letter and invite them to help us to help them.

Z88 Operating Tips

In August, my Son Greg took a liking to my Z88. Greg is in his final semester and will graduate this fall with degrees in Finance and Business Management (Florida State University). Anybody out there need a good man? Now Greg already had the IBM AT computer that I've been knocking for the past couple of years. The IBM is set up in our Motor Home near the college, which is 100 miles East. Maybe I'll get the MH back in time to go to the Milwaukee Computer Fest? So, I made a deal with #1 Son. He can keep the Z88 for the semester if he'll contribute some "HOW TO" articles for Update Magazine. To start him off right I assembled a manual of sorts gleaned from EPROM, the defunct "News Notes", and from Mike Fink (Domino Cubes). The assemblage made a good addition to the Z88 manual which comes up short is some respects.

Greg suprised me the second day. He had worked up a resume with the Z88 and was ready to print it. He came in and used the Update Daisy Wheel printer to print the resume. The print-out was formatted well and looked nice. Not bad for two days usage of a new computer, eh? Then he left for college the next day. His article follows.

WORD PERFECT POWER WITH THE Z88

By: Greg Jones

Sounded too good to be true, so I followed the directions given by Bob Curry, and those given in the Manual for "PC LINK". It works like a charm! I was able to begin a document at home with the Z88, then when I returned to Tallahasse, the document was fed into my IBM with PC LINK, and the finishing touches was done with the IBM software WP-50. The power of Word Perfect's thesarus, spell checker, and formatting, became available with the IBM. Freedom of movement is available with the Z88. The combination of Portability and the extra power of Word Perfect is hard to beat. Then there is the unlimited file storage of the IBM's hard disk to add to the flexibility. Following is a step by step guide for how to transfer files back and forth between the Z88 and the IBM.

1. Create your text with the Z88. It is best to start with something short.
2. SAVE your file to the Z88 Ram Memory, using Menu *3, then arrow to SAVE and NAME your Text.
3. Now you are ready to connect up the two computers.
   a. Key INDEX on the Z88
   b. Insert PC LINK EPROM. Close Cartridge door. Connect Z88 and PC with cable.
   c. Arrow Down to LINK
4. Get into your PC-LINK directory with your hard Drive (The PC LINK Floppy disk was saved to the hard drive previously.) Type PCLINK. You will then be prompted to run your Link Program with the Z88. Key the Link program in the Index Directory of the Z88. This assumes that you have already installed PC-LINK as is

directed by page 2-4 of the Link Guide.
5. Hit RETURN on your PC Computer.
6. Screen #1 found on page 10 of the guide will appear.
7. QUIT Z88 Devices Z88 Dirs Z88 Files
8. HELP PC Devices PC Dirs PC Files

I am using a 128K Ram Cartridge in slot #1, so I change my default devices to RAM .01. The Z88 Dirs is simply your default directory, which in my case is Ram .01. The PC Devices in my case is my hard drive and my PC dirs is my PC-LINK Directory. As I want to send this file to my PC Computer, I select Z88 Files at the first menu.

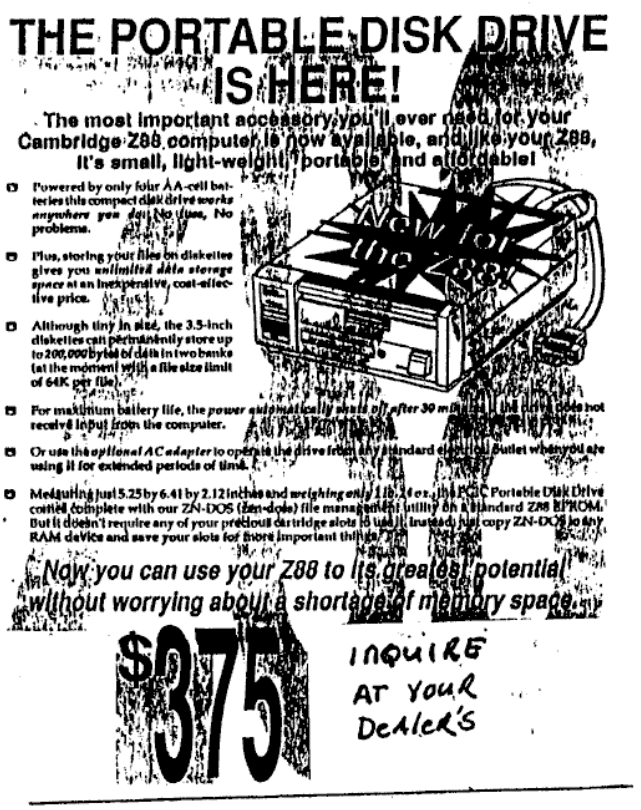
Then I am prompted "Send to PC" Key RETURN.
Program shows me: C:\pclink\filename Key RETURN.
Then there is a little graphics display that shows that your file is actually transferring to the PC.
Once finished you get out of the PC-LINK Program.
9. Now you transfer the new file to WordStar Format. On your PC, type PTW. This is the converter program. You will be prompted to name the file type and the file name, with extension. Then further prompt for "Name want to convert to?". Call it something slightly different and hit RETURN. Now transfer into your Word Perfect program.
10. a. Key Shift + F10, then give path name, plus the file name. In my case this is: C:\PCLINK\file name, and hit RETURN.

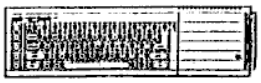
b. Shift + F8; 1 : 7 ; then change both margins to .5 and key RETURN until you get back to text. then hit page down and the document that your started in the Z88 PipeDream will be ready to finish using your Word Perfect Program.

The procedure to get back to PipeDream is not so long.

1. SAVE as DOS Text by using Control + F5.
2. Get out of Word Perfect and back to your PC-Link Directory. Make sure that the two computers are hooked up correctly. Then type PCLINK. The program will prompt you to make sure that your Link program is running.
3. Make sure that your Z88 Devices are still correct. (Mine went back to RAM 0, and I wanted RAM .01.)
4. Select PC-Files. Highlight the file wanted. You will be prompted to send to the Z88, and once you hit RETURN the graphics display will show that the file is transferring.
5. Go to the Menu Files Option and load the newly sent file. In my case the file loads but I can only see the first fifth of a sentence. I used the star + W option and set column to 72, and then I could see the entire document. There may be a better way but this works for me.

So far I have not found a way to get a spread sheet to transfer. I am using TWIN, which is advertised as being 100% compatable with LOTUS. When I try to use LOTUS to Pipe Dream, it kicks back "NOT a Valid Lotus Program". If anyone has an answer to this, I would welcome the information. Greg Jones.


THE PORTABLE DISK DRIVE
IS HERE!
The most important accessory you'll ever need for your Cambridge Z88 computer is now available, and, like your Z88, it's small, light-weight, portable and affordable!
Powered by only four AA-cell batteries this compact disk drive works anywhere you do! No fuss, No problems.
Plus, storing your files on diskettes gives you unlimited data storage space at an inexpensive, cost-effective price.
Although tiny in size, the 3.5-inch diskettes can permanently store up to 200,000 bytes of data in two banks (at the moment with a file size limit of 64K per file).
For maximum battery life, the power automatically shuts off after 30 minutes if the drive does not receive input from the computer.
Or use the optional AC adapter to operate the drive from any standard electrical outlet when you are using it for extended periods of time.
Measuring just 5.25 by 6.41 by 2.12 inches and weighing only 1 lb. 14 oz., the PCC Portable Disk Drive comes complete with our ZN-DOS (zen-dos) file management utility on a standard Z88 EPROM. But it doesn't require any of your precious cartridge slots to use. Instead, just copy ZN-DOS to any RAM device and save your slots for more important things.
Now you can use your Z88 to its greatest potential without worrying about a shortage of memory space.
NOW for the Z88!
$375
INQUIRE AT YOUR DEALER'S



QL+MSDOS

Dick F. Wagner

The ability to access the many MSDOS programs is enticing. A look thru catalogs of 3 West Coast dealers in MSDOS Shareware programs certainly proves the point of the availability of a vast range of programs at very reasonable prices. In quantities of less than 10 disks the prices run about $3.25-$4.00. Some large programs require 2-3 disks. Remember, Shareware involves additional payment in some cases if the program is satisfactory, but not an excessive surcharge (the original payment is to the company, none goes to the programmer).

The catalogs provide good descriptions and sometimes indicate the additional amount of memory required for an IBM PC type computer.

Having this information, is an IBM PC MSDOS type program practical in an emulator form? Once the cost of additional memory, disk driver hardware, a printer interface, and some kind of a "tool kit" program is written off, then the investment is for a program to emulate MSDOS, plus MSDOS itself if it is not in your mix of programs. The cost seems high if MSDOS must be purchased. The "chocalate" version includes version 4.0 and a set of manuals, along with the basic emulator, "SOLUTION" and its manual. The "vanilla" version is the emulator and manual only.

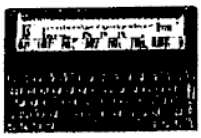
It should be noted that the emulator manual makes no effort to discuss MSDOS programming, except enough to get the MSDOS program loaded. The user must get manuals, books, school classes, etc. to actually use MSDOS. Provision is made in "SOLUTION" to use the dedicated F1-F10 common to MSDOS by providing other key stroke combinations.

The user must have additional memory for the QL as the first added bank of 256K will just get by with MSDOS plus a single disk program if not too long. When considering longer programs like spreadsheets and wordprocessors, a second bank of 256K will be needed unless working with really elaborate programs in which case a fully populated board with 768K would be required.

An MSDOS emulator program makes the mother computer (QL for instance) think it is an IBM PC type computer. It takes two programs to accomplish this emulation. The first, called "SOLUTION" by Digital Precision, is loaded into the computer memory. It does all of the converting from the IBM PC CPU type commands to the QL CPU Superbasic type commands. The second program is MSDOS itself. It is kind of like running 2 programs in series back and forth.

Because of the reduced speed of an emulator (about 1/10 of a PC) I am using an accelerator called "LIGHTNING" which almost gives normal PC speed. My procedure for getting MSDOS into operation is to load "LIGHTNING", then the tool kit (TK2) is initialized, next "SOLUTION" is loaded, and finally MSDOS.

There seem to be several emulators on the English market. Digital Precision has had experience with a CP/M emulator, plus



one for the Atari computers. Reviews in English magazines seem to show a preferance for the DP system as being the most complete and reliable. Sources in the US include RMG Enterprises, 1419 1/2 7th St. Oregon City, OR 97045, and SHARP'S Inc. Box 326, Mechanicsville, VA 23111.

So far my biggest objection to the operation of "SOLUTION" is the need, when in MSDOS, to start all over again with a computer reset when a MSDOS type program hangs up. Maybe that will be reduced with experience with the system and maybe there is something in the "SOLUTION" manual or in the MSDOS manual that gives a clue for solving this. As usual, there is an extended learning process required when switching computer operating systems.

QL TO Z88 LINKUP

Bill Woodward

The QZ comes with all instructions on a mdv cartridge and a 10 foot connecting cable. The cable has two different style end D rings, so only one style can fit in the Z88's D ring. The larger one goes in the QL's serial port. The instructions(contained in README2_doc) say Ser2, but my cable package had Ser1 printed on it. First BUG: the QZ would not transfer with the plugin in either port. I found the answer in P Panel(change defaults etc) on the MENU. I change the default SER2 to SER1 and the QZ performed its act well. When I have a problem I go to the instructions first and to the P Panel for the default devices. All of my problems have resulted from my one tract mind.

The program can be backed-up to either MDV or FLP. Unless you know how to combine the RESPR's of the programs QZ cannot be multitasked with another program. Before you run the program hard copy the instructions by loading README2 into Quill and printing. Then you can run the program turn on or reset. The instructions and on screen MENU OPTIONS are easily understood. Even I was able to transfer a file from Z88 to QL disks after a small difficulty.

The MENU OPTIONS are:

S Send a file from QL to Z88
R Read a file from Z88 to QL
B Batch receive files from the Z88
D Directory of a device
E Send endbatch code
F Rename a file
K Kill a file
V View a file
P Panel (change default etc)
Q Quit to basic
T Translate carriage returns
Transferring a Quill document to Pipedream
Transferring a Pipedream document to Quill

The transferring of a Pipedream document to Quill will let you use the Z88 portable and when you return to your homebase you can transfer the document to Quill and continue to work on it on the large screen of your monitor.
If you are using a serial printer, the QL serial cord will not work, but the Z88 printer cord will let you plug into the serial printer and print out with out going through Quill.

Printing and Formating

Commanding the Printer

This is one of many selected Daisy.B6 program routines that will be excerpted and given to Public Domain. These routines will be selected for their generic value. This one provides PRINTER CONTROL. The need was perceived to Command the printer to CHANGE PRINT STYLES to do a LARGE CENTERED CAPTION, then to RESET to the original style. But it does more that that.

We enter the routines with some established Variables. PS=1=Pica Style. PS=2=Elite Style. PS=3=Micron Style. These are sometimes referred to a Pitch. LX=Max Printer Line for the PS style. QQ=computed Center of Page. TB=established TAB. The Printer Control codes (commands) used are for EPSON Compatable Printers. There are three Sub-routines. <GO SUB 2095> prints a CENTERED CAPTION. <GOSUB 2416 Changes the printer to ELITE DOUBLE HEIGHT CHARACTERS and prints a BILLBOARD as seen in the caption above this column. This routine also uses the next one. <GO SUB 2420> SETS a Print Style. Example: If you first type <LET PS=1> then GOSUB 2420, then the Printer is SET to PICA STYLE. The subroutine first CANCELLS Double Hi characters.

These routines, while in TS-2068 Basic, can easily be cloned for the Sinclair QL and the Z88. In fact, Quill and Pipedream both need such automatic printer control utilities.

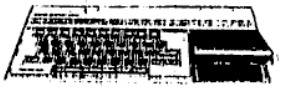
<GO SUB 2416> does the following by line number: 2416 sets aside the page center var "QQ": sets aside the control var "GG": RESETS Var QQ for Page Center for ELITE style: SETS PRINTER to ELITE, then to DOUBLE HI: Prints the caption (M$) by GO SUB 2095: RESETS var QQ and GG. It is to be noted that while the printer's style was changed, the style variable "PS" has not been changed. Therefore as the program moves on to line 2420 the PS= variable still represents the style of print that was in effect.

Line 2020 begins a dual purpose routine. First to reset the original print style after the 2416 routine has been used. OR, <GO SUB 2420 will set any print style represented by "PS= 1 through 3". Explanation: Line 2020 first CANCELLS DOUBLE HI. Then if printing in the COLUMN PRINT MODE, cp is greater than 1, so Micron Print style is set to PS=3.

Print and Format Utilities

```
2095 LPRINT TAB qq-LEN m$/2;m$: LET nn=nn+1
2096 IF gg=1 THEN  RETURN
2097 GO TO fm
2413 IF IN 127=253 THEN  GO TO 2413
2414 RETURN
2416 LET a=2413: LET ga=gg: LET gg=1: LET qx=qq:
 LET qq=(qq*96/120 AND ps=3)+(qq*96/80 AND ps=1)
+(qq AND ps=2): GO SUB a: OUT 127,27: GO SUB a:
OUT 127,120: GO SUB a: OUT 127,1: GO SUB a: OUT
127,27: GO SUB a: OUT 127,77: GO SUB a: OUT 127,
27: GO SUB a: OUT 127,119: GO SUB a: OUT 127,1:
GO SUB 2095: LET qq=qx: LET gg=ga
2420 LET a=2413: GO SUB a: OUT 127,27: GO SUB a:
 OUT 127,119: GO SUB a: OUT 127,0: IF cp>0 THEN
 LET ps=3
2422 IF ps=1 THEN  GO SUB a: OUT 127,27: GO SUB
a: OUT 127,80
2424 IF ps=3 THEN  GO SUB a: OUT 127,27: GO SUB
a: OUT 127,103
2425 IF ps=2 THEN  GO SUB a: OUT 127,27: GO SUB
a: OUT 127,77
2426 GO SUB a: OUT 127,27: GO SUB a: OUT 127,120
: GO SUB a: OUT 127,2: IF cp<1 THEN  LET ll=(INT
 (80*ll/lx+.5) AND ps=1)+(INT (96*ll/lx+.5) AND
ps=2)+(INT (120*ll/lx+.5) AND ps=3)
2427 LET lx=(80 AND ps=1)+(96 AND ps=2)+(120 AND
 ps=3)
2428 IF cp<1 THEN  LET tb=INT ((lx-ll)/2+.5): LE
T qq=INT (lx/2+.5)
2430 IF cp>0 THEN  LET tb=(8 AND cp<2)+(62 AND c
p>1): LET qq=(34 AND cp<2)+(87 AND cp>1): LET ll
=50
2434 RETURN
2436 REM ** For the QL, the Basic Interpreter
doesn't accept some of the BOOLEAN Logic expres-
ions that are in the above lines and the lines
will have to be broken down into QL IF THEN
LOGIC. "BOOLEAN LOGIC FOR QL PROGRAMMING" would
make an interesting Article-  Any Volenteers?
```

Lines 2422 thru 2425 SETS a print style, PS=1=Pica, PS=2=ELITE, PS=3=Micron (pitch). Line #2427 SETS Letter Quality Mode, then Computes a Line Length (LL) that Matches the length of line that was previously in use with the previous print style. Then the Maximum Printer line (LX) is assigned for the current PS=print style. LINE



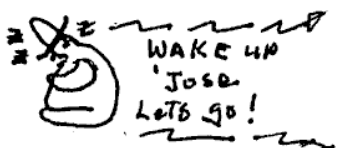
#2428 assigns a new TAB (TB) that will result in the new style's print margins being aligned with the previously printed lines. At this point if Column printing is NOT in effect, the job is done.

Line 2430 works if Column Printing is in effect (CP is Greater than Zero). First Left tab is set for either Left column (CP=1) or Right Column (CP=2). Then Column Center (QQ) is set for either Left or Right Column. Then the Line Length is set to 50.

USAGE: <LET M$="A Caption"> <GO SUB 2095> Prints a Centered Caption. <GO SUB 2416> Changes to ELITE DOUBLE HI, and then prints a Billboard Caption. <LET PS=1 thru 3> <GO SUB 2420> Changes the printer setting to the Style of "PS=", computes MATCHING Line for the new style of print.

Line "BU" is line 2413 which checks the printers status. If the printer is BUSY then <IN 127=253> and the program continues to LOOP Line 2413 until the status changes to READY.

This utility is donated to Public Domain by Update Magazine.



1989 SINCLAIR COMPUTER FEST

We are sponsoring the 1990 Sinclair Fest in Milwaukee. The Date will be June 1 through 3, 1990. That is Friday through Sunday and is the first week end in June. Motels will charge a flat rate of $45 per room, no matter how many will occupy the room. Rooms will have two double beds. Roll away beds extra charge. Motel has a nice indoor pool and the food is excellent.

A banquet is featured at 6:30 on Friday night. We are planning to have a variety of displays including Packet Radio. Seminars are wide open and you are invited to head-up one. Vendors are invited to participate. Tables are available. Set aside the first week end of June for an enjoyable Sinclair Fest. The cost is right! Contact: Bill Heberling (414-527-2191). Or Write: SMUG BYTES, 5052 N. 91st St., Milwaukee, WI 53225.

Ed Notes: Two more issues of Update between now and then will bring more from SMUG about the Fest. For those New Englanders, Westerners, and Southerners, Milwaukee is a nice spot to vacation in June. Lotsa unusual attractions, such as the Experimental Aircraft Asso. Museum, plus others for the family. A good place to visit, with down to earth prices!! (CANADIANS ACCEPTED AT PAR)

from PETER HALE EMSOFT

The TI Group in the Boston Computer Society is planning an Orphan Computer Fair in April/May 1989. Although it's in its early stages, we are very excited. The TI Group has had past shows that have been profitable, so there is a track record of successful organization.

For the record, we in the Sinclair Group do not see it as competing with the SMUG efforts since it will largely be for New Englanders.

However, we like the concept. With each passing year the number of persons interested in a Sinclair-only show, will necessarily be fewer. The more bodies who come into a show, the greater the potential to draw vendors with a range of stuff, particularly those in the generic product, liquidation and remainder business, who would not see a one-product show as justifying the time and expense.

There is also the reality that the more people who come, the greater the sense of camaraderie that can lead to a longer life for support of such computers.

I hope that SMUG will consider broadening their concept to include other orphan groups from the Milwaukee area.

QL NUMBERS INTO TEXT

Peter Hale, P.O. Box 8763, Boston, MA 02114

Computers should save time and effort. This program for the QL does both.

It's purpose is to convert a number (entered as numerals) into an equivalent word string. E.G. input 111.11 and the program returns ONE HUNDRED ELEVEN AND 11/100.

The program handles numbers up to 999999.99 and converts them to the equivalent words.

Originally, the program was part of a check writing routine in ARCHIVE. Here, it is converted to SuperBASIC, while retaining the Archive syntax wherever possible.

Enter the listing. Start by entering the procedure **sav** at line 1400. As you enter the listing, occassionally save the work-in-progress by typing **sav <ENTER>** from the keyboard.

Run the program by **lrun flp2_number <ENTER>** or, if the program is in memory, type either **run**, **start** or **ask_no**, followed by **<ENTER>**.

Although there is limited application beyond check/receipt writing for this program, it is interesting to show the differences and similarities between SuperBASIC and ARCHIVE.

When entering the program in the edit mode of ARCHIVE, ignore lines 10 and 20. Enter each procedure (ignoring line numbers).

There are two other differences. Where the program has a line with a REMark in it, this is the ARCHIVE syntax to replace the Super-BASIC line immediately below it.

The other difference is universal. Wherever SuperBASIC uses the ampersand (&), ARCHIVE requires the plus sign (+).

To run the ARCHIVE version type **run <ENTER>**, then **number <ENTER>**. This auto-starts the program by searching out the procedure called start.

The SuperBASIC program version is 'over-written' to preserve the ARCHIVE syntax that will also operate in SuperBASIC.

For example, procedures no_1 (line 400), no_2 (line 600) and no_2a (line 800) could use SuperBASIC's coercion by changing all occurance of **nu$** to **nu** and removing the quotes from the numerals. Also, one line **IF** statements need not end with : **END IF** nor is it necessary to use the Keyword **LET** in SuperBASIC.

The program points up a weakness of coercion and SuperBASIC's handling of floating point numbers in the procedure ask_no at line 100.

In Archive, the statement after the REMark in line 120 quickly converts numerals to text strings with an accuracy of up to sixteen (16) places.

SuperBASIC gets tangled coercing numbers of more than eight (8) significant places (the decimal point counts as a place). Thus, 100000.01 would not be accurate without the slicing and concatenation in line 130.

THE LISTING:

```
10 ask_no: REMark not for ARCHIVE version
20 :
100 DEFine PROCedure ask_no
110  INPUT "Enter number as digits ";n
120  REMark let nn$=str(n,0,2)
130 n1$=INT(n):n2$=INT(((n-INT(n))+5E-3)
    *100):nn$=n1$&"."&n2$
140  convert:ask_no
160 :
170 DEFine PROCedure convert
180  REMark let dp=instr(nn$,"."): let nn=1
190  dp="." INSTR nn$:nn=1
200  IF dp=2:no_1: END IF
210  IF dp=3:no_2: END IF
220  IF dp=4:no_3: END IF
230  IF dp=5:no_4: END IF
240  IF dp=6:no_5: END IF
250  IF dp=7:no_6: END IF
260  LET txt$=" "&txt$&"AND "&
     nn$(dp+1 TO dp+2)&"/100"
270  IF LEN(txt$)<55
280   REMark let txt$=rept("*",54-
      len(txt$))+txt$
290   txt$=FILL$("*",54-LEN(txt$))&txt$
300  END IF
310  PRINT txt$
320 END DEFine convert
330 :
```

```
400 DEFine PROCedure no_1
410  LET nu$=nn$(nn): LET no1$=""
420  IF nu$="0" AND dp=2: LET no1$="ZERO ":
     END IF
430  IF nu$="1": LET no1$="ONE ": END IF
440  IF nu$="2": LET no1$="TWO ": END IF
450  IF nu$="3": LET no1$="THREE ": END IF
460  IF nu$="4": LET no1$="FOUR ": END IF
470  IF nu$="5": LET no1$="FIVE ": END IF
480  IF nu$="6": LET no1$="SIX ": END IF
490  IF nu$="7": LET no1$="SEVEN ": END IF
500  IF nu$="8": LET no1$="EIGHT ": END IF
510  IF nu$="9": LET no1$="NINE ": END IF
520  LET txt$=no1$
530 END DEFine no_1
540 :
600 DEFine PROCedure no_2
610  LET nu$=nn$(nn): LET no2$=""
620  IF nu$="1":no_2a: RETurn : END IF
630  IF nu$="2": LET no2$="TWENTY": END IF
640  IF nu$="3": LET no2$="THIRTY": END IF
650  IF nu$="4": LET no2$="FORTY": END IF
660  IF nu$="5": LET no2$="FIFTY": END IF
670  IF nu$="6": LET no2$="SIXTY": END IF
680  IF nu$="7": LET no2$="SEVENTY": END IF
690  IF nu$="8": LET no2$="EIGHTY": END IF
700  IF nu$="9": LET no2$="NINETY": END IF
710  LET nn=nn+1:no_1
720  IF nu$<>"0" AND nn$(nn-1)<>"0"
730   LET no2$=no2$&"-"
740    ELSE : LET no2$=no2$&" ": END IF
750  LET no2$=no2$&no1$: LET txt$=no2$
760 END DEFine no_2
770 :
800 DEFine PROCedure no_2a
810  LET nn=nn+1: LET nu$=nn$(nn)
820  IF nu$="0":LET no2$="TEN ":END IF
830  IF nu$="1":LET no2$="ELEVEN ":END IF
840  IF nu$="2":LET no2$="TWELVE ":END IF
850  IF nu$="3":LET no2$="THIRTEEN ":END IF
860  IF nu$="4":LET no2$="FOURTEEN ":END IF
870  IF nu$="5":LET no2$="FIFTEEN ":END IF
880  IF nu$="6":LET no2$="SIXTEEN ":END IF
890  IF nu$="7":LET no2$="SEVENTEEN ":END IF
900  IF nu$="8":LET no2$="EIGHTEEN ":END IF
910  IF nu$="9":LET no2$="NINETEEN ":END IF
920  LET txt$=no2$
930 END DEFine no_2a
940 :
1000 DEFine PROCedure no_3
1010  LET nu$=nn$(nn): LET no3$="":no_1
1020  IF no1$<>"": LET no3$=no1$&"HUNDRED "
1030   ELSE : LET no3$="": END IF
1040  LET nn=nn+1:no_2
1050  LET no3$=no3$&no2$: LET txt$=no3$
1060 END DEFine no_3
1070 :
1100 DEFine PROCedure no_4
1110  LET nu$=nn$(nn):no_1
1120  LET no4$=no1$&"THOUSAND "
1130  LET nn=nn+1:no_3
1140  LET no4$=no4$&no3$: LET txt$=no4$
1150 END DEFine no_4
1160 :
1200 DEFine PROCedure no_5
1210  LET nu$=nn$(nn):no_2
1220  LET no5$=no2$&"THOUSAND "
1230  LET nn=nn+1:no_3
1240  LET no5$=no5$&no3$: LET txt$=no5$
1250 END DEFine no_5
1260 :
1300 DEFine PROCedure no_6
1310  LET nu$=nn$(nn):no_3
1320  LET no6$=txt$&"THOUSAND "
1330  LET nn=nn+1:no_3
1340  LET no6$=no6$&no3$: LET txt$=no6$
1350 END DEFine no_6
1360 :
1400 DEFine PROCedure sav
1410  REMark save "flp2_number"
1420  DELETE "flp2_number":
      SAVE "flp2_number"
1430 END DEFine sav
1440 :
1450 DEFine PROCedure start
1460  ask_no
1470 END DEFine start
```

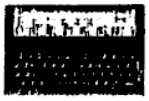
Note: This article is prepared on a 24 pin Panasonic KX-1124 printer in letter quality Courier font, micron (15 char/inch) pitch.

It requires only minor adjustments to Epson control codes through the install_bas program to access the vast majority of the features of this printer under Quill.

<Musings Cont> From Pg 25

SERIOUS BUSINESS! I've been in a quandry for a year about TO or NOT To buy a Hard Drive for my QL. Al Feng's article on Networking turned the trick! I'm gonna. Here is my rationale. My QL's are operating SUPER STABLE, and I do believe that I have more computing power than does one have with an IBM AT, except for the hard drive. The Brit Pound conversion is about 1.55, which means that a Hard Drive at 355 pounds comes to about $550.00 (US). Last year I paid $650.00 for a 40 mgbt hard drive for my Son's IBM, and then had to buy a $175.00 controller for it. The price for the QL Hard Drive is right! Now picture this: Two QL's networking, one with 2 floppies and the

Continued Page 49



PRINTERS and PRINTER CONTROL

The genesis of this work-up goes back to my early work with the Sinclair ZX-81, a computer designed "not to print". Soon after the Z-81 reached the "million mark", Cottage industry developed Machine Code "Printer Drivers". From then until today, our Sinclair Computer fans have not had a minimum diet of "How To Control Printers". That is, other than via using a copyrighted software and following the manual instructions.

CONCEPTS: 1. Printer Driver Software: These Machine Code utilities provide the Software LINK between the Computer, the Centronics Parallel Interface, and the Printer. The "prcode" program usually resides in RAM, but some prcodes are put into EPROM. The code programs work marvelously to send BURSTS of data from the computer to the printer. Such bursts of data are not FORMATTED. Thus, a separate software is needed to control the data bursts so that Afaganistan, is printed in its entirety on the same line of print. Also, and this is the thrust of this article, the USER NEEDS PRINTER CONTROL. User control is needed so that there is a way to make the printer change its STYLE of PRINT and other PRINTER MODES, via software or direct COMMANDS.

ADDITIONAL CONCEPTS: Some Computers, such as the Unexpanded QL and the TS-2068, have little spare FREE MEMORY to store elaborate ancillary programs such as Printer Control utilities. Thus any such utility should be as brief as is practical to do the jobs needed. What are the JOB TASKS that most users need for controlling a printer "that the prcode does not accomplish"? Lets list them: 1. Menu Control for Automatic printer MODE Changes. 2. Line Length Control. 3. Page Length Control. 4. Margins Control. 5. Center Print of Headers Footers. Each of these have branch needs.

I didn't list this one because it is so broad. There are many different kinds of printers. A control program "for just one printer" would have very limited application. We really need a "universal printer control system" that can be used for Dot Matrix printers, Daisy Wheel Printers, and all makes of both. Now that is a tall order, isn't it? AND is is make even taller when one attempts to develop a Control System that works with both the Sinclair QL and the TS-2068. The task must be approached with the knowledge that some "user adjustments" will always be required for different computers and different printers, and different Centronics Interfaces. If the needs for user adjustments are well defined, then such a universal program will be useful to many.

PRINTER CONSIDERATIONS

THE NEW 24 PIN PRINTERS: Recently one of Update's printers, a Star Micronics 1000 dot matrix, reached its 365th day of warranty, and promptly expired. The Print Head quit printing half of the characters, which made English look like gobeleze, a dialect of the Australian aborigines. To get it fixed would have required about $85 skins, which did not seem to be cost effective, what with the "out of the door, out of warranty" situation with repairs. So I opted for a new Panasonic KXP-1124 printer. This is a 24 pin printer, whereas the old Star was a 8 pin printer. Why mention that? Well I have read where there is much confusion as to whether our computers (QL and TS-2068) will even work with a 24 Pin Printer. Lets put that to rest imediately.

THEY WILL, and with GUSTO! AND without any kludge-up at all. If your 8 pin Dot Matrix prints something via "LPRINT", it will do the same, but with three times better print quality than with an 8 pin printer. "3 times better" because the 24 pin printer prints three times as many dots for a character printed. DRAFT quality with the 24 pin printer is better than NLQ quality with most 8 or 9 pin printers. For proof of that, this column is printed with the new 24 pin printer (in Draft Mode).

DAISY WHEEL PRINTERS: The main difference in these and Dot Matrix is the Daisy Wheel has a "print wheel" that has "whole formed characters". A solenoid Whacks the reed that has the "die of the character", and a perfect character prints to paper. The other main difference is that a Daisy Wheel printer is SLOW, about 17 characters per second speed. Daisy Wheel printers are easier to control than dot matrix printers. This is because there are only a few different modes of printing, and my new Panasonic has more than 5000 different possible set ups. Of course, that many set-ups are impossible to program in an automatic Menu Controlled program. So we must select a dozen or so "preferred modes" and use the printer's front panel switches to control the unusual modes.

Universal Variable File, "varset.B6"
Donated to Public Domain by Update Magazine

```
9902 CLEAR : LET a=0: LET b=0: LET c=0: LET xx=0: LET
 a1=0: LET u$="": LET w$="": LET m$="": LET sp=2076::
 FOR r=1 TO 1: NEXT r: LET pe=5898: LET 11=64: LET tb
=8: LET pp=0: LET sd=0: LET nn=0: LET ln=58: LET oa=1
: FOR n=1 TO 1: NEXT n: LET jy=0: DIM h$(1,1): DIM i$
(1,1): LET id=0: LET a4=0: FOR y=1 TO 1: NEXT y: LET
turbo=0: LET a3=0: LET red=0: LET m1=100: DIM v$(1,1)
: LET x=0: LET nt=0: LET nx=0: LET x1=0: LET z2=0: LE
T mm=0: LET lx=80: LET z=0: LET d1=0: LET d2=0: LET a
1t=0: LET ps=2: LET oo=0: LET ob=2: LET oc=3: LET od=
4: LET oe=5: LET of=6: LET og=7: LET oh=8: LET oi=9:
LET oj=10: LET ok=11: LET ol=12: LET om=13: LET on=14

9940 DIM o$(1,1): LET op=15: LET oq=16: LET or=17: LE
T os=18: LET ot=19: LET ou=20: LET ov=21: LET ow=22:
LET ox=23: LET oy=24: LET oz=25: LET xo=27: LET po=12
7: LET tel=3742: LET pc=0: LET az=0: LET no=0: LET cx
=0: LET cz=0: LET eq=0: LET ep=0: LET ik=2554: LET il
=2550: LET k1=1000: LET k2=2000: LET k3=3000: LET k4=
4000: LET k5=5000: LET k6=6000: LET k7=7000: LET k8=8
000: LET k9=9000: LET m2=200: LET m3=300: LET m4=400:
 LET m5=500: LET m6=600: LET m7=700: LET m8=800: LET
m9=900: LET t3=30: LET t4=40: LET t5=50: LET t6=60: L
ET t7=70: LET t8=80: LET t9=90: LET sq=2576: LET seq=
0: LET alt=2070: LET gg=0: LET fm=2070: LET ed=2926:
LET te=3741: LET tel=3742: LET m1=2000: LET rp=2100:
LET fo=2083
9960 LET qs=0: LET sx=oo: LET xy=oo: LET ep=0: LET eq
=0: LET cx=0: LET cy=0: LET pc=0: LET az=0: LET no=0:
 LET z1=0: LET rpr=0: LET pt=0: LET ms=0: LET mx=0: L
ET gg=0: LET lb=0: LET z2=0: LET zy=0: LET sp1=0: LET
 qq=0: LET scp=0: LET mm=0: LET zm=0: LET mz=0: LET a
z=0: LET no=0: LET nx=0: LET z5=0: LET df=0: LET z1=1
: LET z2=1: LET sr=0: LET sw=0: LET sp1=0: LET rpr=0:
 LET lm=0: LET ms=0: LET in=0
9991 POKE 23298,22: POKE 23299,8: LET mt=8031: LET co
m=0: LET bu=8080: LET exp=0: LET it=0: LET und=0: LET
 scrp=0: LET lh=1: LET pg=1: LET cp=0: LET xa=1: LET
q$="": LET r$="": LET s$="": LET t$="": DIM n$(13,31)
: LET n$(6)="UpDate Magazine": LET n$(7)="1317 Stratf
ord Ave.": LET n$(8)="Panama City, FL 32404": LET n$(
9)="904 871 3556": LET n$(11)="Sincerely,": LET n$(12
)="904 871 4513": LET n$(13)="Bill Jones": LET a$="":
 LET b$="": LET c$="": LET d$="": LET e$="": LET f$="
": LET g$="": LET j$="": DIM k$(1): LET l$=""
```

The FOUR KEY IN UTILITIES
The "varset.B6" Program

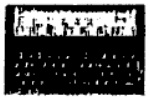
There are four programs to be presented. Each is done for the TS-2068, but each can easily be adapted for the QL, that is, if I can adequately explain the programming rationale. Lets start with the program "varset.B6". This is a program designed to be MERGED to either the TS-2068 or the QL. The program has HIGH LINE numbers that should not interfere with any existing program. All of the variables used in the other three programs are "initialized" by this "varset.B6" program via <GO TO 9902>. ONCE DONE, you can DELETE the program, as the variables are then in memory. Use <DELETE 9902,> and the vars table will stay in memory for your work with the other three programs. This variable file forms the basis for MUCH MEMORY SAVINGS in all programming that you will create.

As a general rule, if your Basic program is over 4K in program length, the use of these variables within the program (instead of raw numbers) will equalize. That is to say, that the memory loss of 1751 bytes for the vars file storage in memory will be recouped. After that, the continued use of the vars instead of raw numbers will multiply in memory savings. A typical program length of 20K will be reduced by about 5000 bytes via the use of the vars for numbers. I use a system that lends itself to easy recollection. Numbers zero through 25 are represented by the "oo" variables, "zero=oo, oa=1,ob=2, through on=14. Skip oo because oo=zero, pick up with on=16, through oz=25".

One can count on one's pinkies at first until their habitual use sinks into the noggin, and one thinks "oj instead of ten". Then "tens" are "t3 through t9". "t3=30, t4=40, etc". Then Hundreds are represented by "M1 through M9", and Thousands are represented by "K1 through K9". Thus <GO TO k2+ob> would be <GO TO 2002>. All other variables in the "varset.B6" file are oft used numbers in the Daisy.B6 Software. These "line numer and program specific vars" may be changed for the needs of a specific program. So much for "varset.B6". Key it in and SAVE it to disk. Once done you can always MERGE it, then <GO TO 9002> to initialize this variable file .

The "fomat.B6" Program

THE FORMAT MENU GROUP, and the PRINT STYLE



The Format Menu Group, "fomat.B6"

Donated to Public Domain by Update Magazine

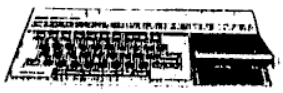
```
2080 RANDOMIZE USR ml: LOAD "fo5.C1"SCREEN$ : LET mat
=11: LET xo=1x: LET lo=11: LET rs=id: LET cp=oo: LET
lh=oo
2081 PRINT AT oi,VAL "24";"id=";id;"  ";AT oe,os;("@"
 AND sd<ob)+(" " AND sd>oa);AT of,os;("@" AND sd>oa)+
(" " AND sd<ob);AT op,ob;("Pica " AND ps<ob)+("Elite"
 AND ps=ob)+("Cond " AND ps>ob);AT ot,ob;1x;AT op,oq;
qq;AT op,VAL "28";ln;AT ot,oi;11;AT ot,oq;mat;AT or,V
AL "28";nn;AT os,VAL "28";pg;AT ot,VAL "28";tb: GO SU
B sq: GO SUB il: GO TO (z=oh)*VAL "2093"+(z>oh)*VAL "
2081"+(z<oa)*VAL "2081"+(z>oo AND z<oh)*VAL "2082"
2082 IF x=og THEN  LET 1x=(VAL "126" AND pr>oa)+(VAL
"120" AND pr<ob): LET 11=(VAL "50" AND pr>oa)+(VAL "4
8" AND pr<ob): CLS : PRINT AT oj,ob;"Start Col 1 or 2
 ?": INPUT "key 1 or 2";cp: IF cp=oa THEN  LET qq=(VA
L "34" AND pr>oa)+(VAL "34 AND pr<ob"): LET tb=oj
2083 IF z=og AND cp=ob THEN  LET qq=(VAL "90" AND pr=
ob)+(VAL "86" AND pr=oa): LET tb=(VAL "64" AND pr=ob)
+(VAL "62" AND pr=oa)
2084 IF z=og THEN  LET ps=oc: IF pr=oa THEN  LET com=
oa: GO SUB VAL "498": LET a=VAL "2413": GO SUB a: OUT
 po,VAL "27": GO SUB a: OUT po,VAL "27": GO SUB a: OU
T po,VAL "120": GO SUB a: OUT po,oa: GO SUB a: OUT po
,VAL "27": GO SUB a: OUT po,VAL "103"
2085 LET sd=(z AND z<oc)+(sd AND z>ob): IF z=od THEN
 LET tb=(tb-oe AND 11<1x-oi)+(tb AND 11>1x-oj): LET 1
1=(11+oj AND 11<1x-oi)+(11 AND 11>1x-oj): LET id=id-o
j: BEEP oa/od,ol: PRINT AT ot,oi;11;AT ot,oq;mat;AT o
t,VAL "28";tb;" ": GO TO VAL "2081"
2086 IF z=oe THEN  LET tb=(tb+oe AND 11>ou)+(tb AND 1
1<VAL "21"): LET 11=(11-oj AND 11>ou)+(11 AND 11<VAL
"21"): LET id=id+oj: BEEP oa/od,ol: PRINT AT ot,oi;11
;AT ot,oq;mat;AT ot,VAL "28";tb;" ": GO TO VAL "2081"

2087 GO TO (z<oc)*VAL "2081"+(z=oc)*VAL "2093"+(z>oc)
*VAL "2088"
2088 BEEP oa/od,os: LET lh=(oa AND z=oh)+(oo AND z<>o
h): IF cp<oa AND pr=oa THEN  LET 1x=(VAL "120" AND ps
>ob)+(VAL "96" AND ps=ob)+(VAL "80" AND ps<ob): REM D
ot Matrix
2089 IF cp<oa AND pr=VAL "2" THEN  LET 1x=(VAL "124"
AND ps>VAL "3")+(VAL "96" AND ps=VAL "3")+(VAL "80" A
ND ps<VAL "3"): LET qq=INT (1x/ob): REM Daisy
2090 CLS  : PRINT AT oj,ob;"INPUT Printer Line Spacin
g"''"<1> 1/8""=76 lines;"''"<2> 11/64""=56 Lines"''"<
3> 3/16""=50 Lines"''"<4> 1/4""=37 Lines": INPUT x: L
ET ln=INT (VAL ".6"+(VAL "9.5"/VAL ".125" AND x<ob)+(
VAL "9.5"/VAL ".172" AND x=ob)+(VAL "9.5"/VAL ".187"
AND x=oc)+(VAL "9.5"/VAL ".25" AND x>oc)): CLS : PRIN
T AT oj,ob;"Set Printer for Line Spacing"''TAB ob;"Th
```

MENU GROUP: These two programs work hand in glove to provide a miriad of PRINTER CONTROL and FORMATTING needs. We will first discuss the FORMAT MENU GROUP, which is designed to provide Automatic Format assistance. The Menu Itself is reproduced below. Incidentally, this menu is produced "right within the text", using the Daisy.B6 software for the TS-2068.

```
   Select until Finished

  <1> Sngle Spc
  <2> Dbl Spc
  <3> Print Style Menu
  <4> Block Indent Off

  <5> Block Indent ON
  <6> To Fun Menu
  <7> Column Print
  <8> Let Head Stnry
                          2083
      Pg Cent=     Pg Lgth=

Max     Line    Match  Line=
Line    Lgth    Line   Page=
                       Tab =
```

For cross reference purposes, you can refer to the above Menu, its numbered selections, and the LLIST of ""fomat.B6"". The menu is a "rotary", in that all selections "1 through 5" return the menu after the selection task is done. These selections cause a graphics block to be posted to show the new status. The current Format Status is displayed with their vars identification. This makes a pretty busy screen display, but one that keeps you oriented as to the format status.

When referring to the LLIST, there is one odd character that is wrong in many places. The "@" in the listing is actually a graphics block, which is posted at various places on screen to designate current status. When keying in use the graphics block for this character. The menu provides many electives for Formatting. Some electives follow via prompts after the selection is made to return to the FUNCTION MENU. Basically, the menu electives cause calculations for using Format Line Length, TAB, and Center of Page, to format CENTER PRINTED LINES on a page. We start with the premise that the printer itself is DUMB, and is set to its start up initialization parameters. NO



MARGINS, no Porportional Print, and the printer is set for the usual 60 lines per page (most initialize that way).

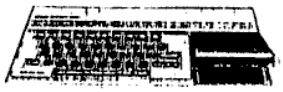
Two special set-ups are provided. <LETTER HEAD Stationary> selection sets variable "lh" to be used by a word processor. The Daisy program that uses this group of menus uses var "lh" to roll up 12 lines before printing the Business Block of a letter. This lets one use pre-printed letterhead stationary. Then if a second page prints, it begins at the regular place on page 1" from the top of form. <COLUMN PRINT> selection results in a prompt "Start at Col 1 or Col 2?", then "Start at Line #?". This allows one to start column printing anywhere with either the left column or the right column.

When in Column Print Mode, the line length and left margin are set for fixed values. (50 chrs per line for Dot Matrix, or 52 for Daisy Wheel printers). You can change this in line number 2082. The TAB "tb" and Center of page "qq" may also be changed for your Condensed print line length. One should have CONDENSED PRINT set up either manually or via the Style Menu to use the Column Print mode. The PAGE END utility "pe.B6" is used by the Dasiy.B6 program for Switching columns in mid print stream. The text is printed beginning at the column selected, and continues printing alternate columns until all data has been printed. This works better with Cut Sheets, as you can re-insert the page at each page end to let the next column print on the same page.

This Panasonic dot matrix printer allows Paper Parking, and Cut Sheets to be fed into the printer from the front. This way, I can print alternate columns on the same page. Also, to be experimented with is a capability to "reverse direction line feed". Perhaps I can develop a little routine for fan fold forms to "back up to the top of form" and continue column print on the same fan fold page? In discussing these functions, the Daisy software is used as an example of how the miriad of Formatting functions can be used by a software, using the array of FORMATTING VARIABLES that are established at this format menu. In a later issue of Update we will give the "LINE FORMATTER-PRINTER" which makes use of the printing variables that this "fomat.B6" program establishes. For now, you can see the results in these printed columns.

When <FUNCTION MENU> is the selection, a

```
en Enter Lines per page?"''TAB ob;"Current page lgth=
";ln: INPUT ln: CLS : INPUT "Enter Page#";pg: INPUT "
Enter Line Count start";nn: IF cp>oo THEN  GO TO VAL
"2092"
2091 CLS : PRINT AT oj,oo;"Line length=";ll''"Match L
ine=";mat'': INPUT "New Line Length?";ll: LET qq=INT
(lx/ob+oa/ob): LET tb=INT ((lx-ll)/ob+oa/ob): PRINT "
Tab=";tb: INPUT "New Tab?";tb


The Style Menu Group, "stynn.B6"

Donated to Public Domain by Update Magazine

8001 LET it=oo: LET com=oo: LET exp=oo: LET und=oo: L
ET scrp=oo
8002 RANDOMIZE USR ml: LOAD "ps.C1"SCREEN$ : LET c=(o
f AND ps<ob)+(og AND ps=ob)+(oh AND ps>ob): PRINT AT
of,ob;" ";AT og,ob;" ";AT oh,ob;" ";AT c,ob;"@";AT os
,og;lx;" ";AT os,op;tb;" ";AT ot,og;qq;" ";AT ot,op;l
l;" ";AT oj,ob;(" " AND com=oo)+("@" AND com=oa);AT o
l,ob;(" " AND scrp=oo)+("@" AND scrp=oa);AT om,ob;("
" AND exp=oo)+("@" AND exp=oa);AT on,ob;(" " AND it=o
o)+("@" AND it=oa);AT op,ob;(" " AND und=oo)+("@" AND
 und=oa): LET px=ps: LET xo=lx: LET px=ps: LET lo=ll
8003 GO SUB sq: GO SUB il: IF z<oa OR z>oi THEN  GO T
O VAL "8003"
8004 IF z=oi THEN  GO TO VAL "8114"
8005 IF pr>oa THEN  LET lx=(84 AND ps<ob)+(VAL "98" A
ND ps=ob)+(VAL "126" AND ps>ob): LET qq=INT (lx/ob+oa
/ob)
8006 IF pr=oa THEN  LET lx=(VAL "80" AND ps<ob)+(VAL
"96" AND ps=ob)+(VAL "120" AND ps>ob): LET qq=INT (lx
/ob+oa/ob)
8010 IF pr=oa AND z=od THEN  LET com=oa: GO TO VAL "8
002"
8012 IF pr=oa AND z=oe THEN  LET scrp=oa: GO TO VAL "
8002"
8014 IF pr=oa AND z=of THEN  LET exp=oa: GO TO VAL "8
002"
8016 IF pr=oa AND z=og THEN  LET it=oa: GO TO VAL "80
02"
8018 IF pr=oa AND z=oh THEN  LET und=oa: GO TO VAL "8
002"
8020 IF z<od THEN  LET ps=z
8021 IF pr=ob THEN  LET lx=(VAL "84" AND ps=oa)+(VAL
"98" AND ps=ob)+(VAL "124" AND ps=oc): LET qq=INT (lx
/ob+oa/ob): GO SUB VAL "8026": GO TO VAL "8002"
8024 CLS : PRINT AT oj,oc;"One Moment Please-"''TAB o
c;"Printer is setting. . .": LET lx=(VAL "80" AND ps=
oa)+(VAL "96" AND ps=ob)+(VAL "120" AND ps=oc): LET q
q=INT (lx/ob+oa/ob): GO SUB mt: GO SUB VAL "8050"
8026 LET ma=INT (oa/ob+lx*lo/xo): GO SUB VAL "8028":
BEEP oa/ob,ov: GO TO VAL "8002"
8028 CLS : PRINT AT oj,ob;"Key in Line Length"''TAB o
```



```
b;lx;"=Max   ";ma;"=Match Margins": INPUT "Input Line
 Length ";ll: LET tb=INT ((lx-ll)/ob+oa/ob): CLS : RE
TURN
8030 REM ** The Dot Matrix                    Printer C
ontrol **
8031 IF pr<>oa THEN  RETURN
8034 GO SUB bu: OUT po,VAL "18": GO SUB bu: OUT po,VA
L "27": GO SUB bu: OUT po,VAL "119": GO SUB bu: OUT p
o,oo: GO SUB bu: OUT po,VAL "27": GO SUB bu: OUT po,V
AL "53": GO SUB bu: OUT po,VAL "27": GO SUB bu: OUT p
o,VAL "45": GO SUB bu: OUT po,oo
8036 GO SUB bu: OUT po,VAL "27": GO SUB bu: OUT po,VA
L "120": GO SUB bu: OUT po,oa: GO SUB bu: OUT po,VAL
"27": GO SUB bu: OUT po,VAL "107": GO SUB bu: OUT po,
(oo AND ps<oc)+(ob AND ps>ob)
8038 GO SUB bu: OUT po,VAL "27": GO SUB bu: IF ps=oa
THEN  OUT po,t8
8040 IF ps=ob THEN  OUT po,VAL "77"
8042 IF ps=oc THEN  OUT po,VAL "103"
8044 RETURN
8050 IF com=oa THEN  LET lx=(VAL "137" AND ps<ob)+(VA
L "160" AND ps=ob)+(VAL "120" AND ps=oc): LET qq=INT
(lx/ob+oa/ob): GO SUB bu: OUT po,VAL "27": GO SUB bu:
 OUT po,VAL "15"
8052 IF exp=oa THEN  LPRINT : LET nn=nn+oa: GO SUB bu
: OUT po,VAL "27": GO SUB bu: OUT po,VAL "119": GO SU
B bu: OUT po,oa
8054 IF it=oa THEN  GO SUB bu: OUT po,VAL "27": OUT p
o,VAL "52"
8056 IF und=oa THEN  GO SUB bu: OUT po,VAL "27": GO S
UB bu: OUT po,VAL "45": GO SUB bu: OUT po,oa
8058 IF scrp=oa THEN  GO SUB bu: OUT po,VAL "27": GO
SUB bu: OUT po,VAL "120": GO SUB bu: OUT po,oa: GO SU
B bu: OUT po,VAL "27": GO SUB bu: OUT po,VAL "107": G
O SUB bu: OUT po,od:
8060 RETURN
8080 IF IN 127=253 THEN  GO TO bu
8082 RETURN
```

"pe.B6", the column switching Page End Routine

Donated to Public Domain by Update Magazine

```
5900 IF pr=oo THEN  LPRINT '''''';TAB qq;pg''''''''
'': LET nn=oa: LET pg=pg+oa: RETURN
5901 IF nn<ln+ob THEN  LPRINT : LET nn=nn+oa: IF nn
<ln+ob THEN  GO TO VAL "5901"
5902 IF cp>oo THEN  GO TO VAL "5906"
5903 LPRINT TAB qq;pg: LET nn=oo: LET pg=pg+oa: OUT
 po,ol: IF pr=oa THEN  GO TO VAL "5913"
5904 GO SUB VAL "5912": GO TO VAL "5913"
5906 IF cp=ob THEN  LPRINT TAB lx/ob;pg: LET pg=pg+
oa
5908 OUT po,ol: LET nn=oo: IF pr>oa THEN  GO SUB VA
L "5912"
```

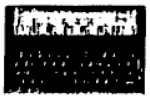
series of prompts appear to allow "Exceptional" Formats. These allow one to input different TAB and LINE LENGTH to achieve OFF-SET printing. The printing variables are normally set for text printing to be centered on page, or in pre-set columns. These prompts offer the choices of other formats. Also, a pair of prompts allow for condensing or expanding the increments of spacing between lines, to produce such printing as subscript or superscript, or Daisy Wheel characters that have reduced or expanded height. For example, one use that I make of these prompts is to use a print wheel having reduced height charcters that produce 76 lines of characters on a page, instead of a normal page of 56 lines.

The "fomat.B6" program allows for either Daisy Wheel or Dot Matrix printers that have different "maximum columns per line". This differentiation is detected by variable "pr=1 for Dot Matrix" or "pr=2 for Daisy Wheel". This also is a nicety of the Daisy.B6 software. This Format program is coordinated with the next program "stymn.B6" which provides for both Daisy Wheel and Dot Matrix Printer control.

The Format program works on "an existing print style" to allow a complete range of formats to be set up. But in one instance it does actually change a print style. That is when "Column Print" is elected. Then the printer is commanded to change to "Micron pitch", which for some printers is called "Condensed". This produces 120 Character line (maximum. Column print is automatically set for 50 character lines for this printer.

USING MERGE/DELETE

THE MERGE CONCEPT: All of these programs are designed to be MERGED to any operating Program, and once used to be DELETED. A typical program line to MERGE the "fomat.B6" program is given in TS-2068 LKDOS Syntax: <2079 RANDOMIZE USR 100: MERGE "fomat.B6">. Then anywhere within the host program, <GO TO 2079> will MERGE the fomat.B6 program and it will operate. A typical DELETE line (after the fomat.B6 program has been used) would be: <2093 DELETE 2080,2091: GO TO fm>. These two lines are exactly the ones used in the Daisy software to Merge in the Format Menu and when finished with it to delete it. Note the menu routing in line 2081 in which if <8 Format Menu is elected> the program is DELETED by line 2083. Since you only need such menus to Set Up Parameters Ocasionally, it shouldn't be stuck in



memory to sacrifice data capacity.

The PAGE END PROGRAM

The PAGE END ROUTINE, "pe.B6": This utility is given only because it relates to the "Column Print Mode" as set up by the Format Menu. This program ALSO is designed to work with either Daisy Wheel or Dot Matrix Printers. If using a Daisy Wheel, a prompt is given to "Service Printer", which means, "Feed another Sheet, or Re-insert the Same Sheet". This allows the alternate Column to be printed on the same page. For Dot Matrix printers (pr=1), the pause is not given between pages. The variable "cp" value of 1 or 2 is used to switch to the alternate column, then re-set the TAB "tb" for printing the next column. Again, the differences between print size between a Dot Matrix and a Daisy Wheel printer is accounted for in the calculations. Ordinary printing with "page width" columns bypasses the "column switcher" because "cp=0".

THE STYLE MENU GROUP

```
Select Print Style

<1> Pica  -80-     Com=137
<2> Elite -96-     Com=160
<3> Micron-120-    same

<4> Com  137/160   ]  †
                   ]
<5> Script         ] Set
<6> Expand Height  ] First
<7> Italics        ]
<8> Underline      ]
<9> Format Menu

LX=80   TB=15
QQ=40   LL=50        8000
```

The program "stymn.B6" is the PRINT STYLE MENU and the PRINTER CONTROL PROGRAM. This program is also designed to be MERGED as needed, and when it is used, to be DELETED from memory. The program line to MERGE "stymn.B6" is given in TS-2068 LKDOS syntax to be emulated by other DOS Syntax. It is: <8000 RANDOMIZE USR 100: MERGE "stymn.B6">. Then the DELETE LINE is: <8114 DELETE 8001,8082: GO TO fm>. "fm" is line #2070 of the Daisy software (is Function Menu).

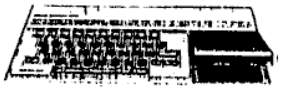
```
5909 LET cp=(oa AND cp=ob)+(ob AND cp=oa): IF cp=oa
 THEN  LET qq=(VAL "34" AND pr=ob)+(VAL "34" AND pr
=oa): LET tb=oh
5910 IF cp=ob THEN  LET qq=(VAL "92" AND pr=ob)+(VA
L "86" AND pr=oa): LET tb=(VAL "66" AND pr=ob)+(VAL
 "62" AND pr=oa)
5911 GO TO VAL "5913"
5912 BEEP oa,ov: CLS : PRINT AT oj,ob;"Feed New She
et, then ENTER": PAUSE oo: LET nn=oo: CLS : RETURN
```

```
The MERGE LINE:  5899 PRINT #4: MERGE "pe.B6"
The DELETE LINE:  5914 DELETE 5900,5912: RETURN
```

"stymn.B6" first defines the "font variables" in line #8001. They are: "it=1 for Italics", com=1 for Compressed print, exp=1 for EXPAND Hight, und=1 for Underline ON, scrp=1 for Script style ON. These are control variables that are set by the menu electives 4 through 8. They are not "Print Sizes". The first three 1-3 electives are "Print Size or Style". The menu is a "rotary" in that the selections 4 through 8 are used first to elect the fonts. None of these affect the actual width of the characters,except "4 Compress", which has different results depending upon the basic print style in use. Once the fonts have been elected, and up to four may be combined, the last selection is the Print Style (1 through 3).

The selections 1-3 selects the Print Style. Line #8020 is reached only by selecting one or the three basic Print Styles (1-3). This line assigns the Print Style variable "ps=1 through 3". Then lines 8021 through 8028 are executed. Twin routines are for either Dot Matrix or Daisy Wheel printers. These are differentiated by the variable "pr=1 for Dot Matrix, or pr=2 for Daisy Wheel". Other operating vars are "ll=Line Length, qq=page center, tb=TAB". We enter the program at line 8001 with these printing vars having been in use or set by the software initialization. Then at the last part of line #8002, these printing vars are "set aside to be used for calculations" when the style is changed at menu.

These vars values are set aside (in line #8002) by asigning their "past values" to "px=ps, xo=lx, and lo=ll. Then px and ps= the previous print style 1-3, xo and lx= the previous Max Printer line for the past print style in use. "lo



and 11"= the past Line Length in use. The calculations beginning at line 8021 computes a "MATCHING Set of Parameters" so that you can elect a MATCHING Line length for a new print style selected at menu. The Matching Line length is presented on screen with a prompt "Input Line Length". Then computations are done to FORMAT THAT Line Length to be centered on page. A new TAB "tb" and Page Center "qq" is calculated and assigned. Now we have Line Length "ll", page center "qq", and TAB "tb" assigned for the newly selected print style.

At this time, if Daisy Wheel printer is in use (pr=2), the Style Menu is re-called, where an escape is provided back to the FORMAT Menu. The Daisy Wheel printer requires no further commanding, because it has a fixed print wheel. Why then do we even use the Style menu for a Daisy Wheel? Well, it is quick and easy to change print wheels, and this program will match up the printing with a new print wheel's style of print.

DOT MATRIX PRINTER CONTROL

The DOT MATRIX PRINTER CONTROL Section begins at line #8034. After all of the calculations has been done and the Format for the New Style of print has been set up, an obscure command in line #8024 is "GOSUB mt", followed by "GOSUB 8050". "Line mt" is 8031. We will now SET THE PRINTER to the newly selected style and fonts. This section deserves the most explanation because it is where most users will bave to work on to Match the Printer Control Commands to their specific printer. It really is quite SIMPLE, and YOU CAN DO IT!

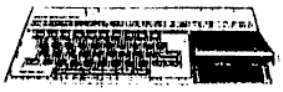
THE PRINTER CONTROL GROUP FOR A PANASONIC KX-P1124 24 pin Dot Matrix Printer: We begin at LINE #8031 which rejects anything BUT a Dot Matrix Printer. Then the Dot Matrix Control section begins at line #8034. The very first task to perform is to RESET the Printer to its TURN ON STATUS- That is EXCEPT for the TOP OF FORM setting. We do this by installing the proper "CANCEL COMMANDS" for every special Font and Style that we use. This cancels the printer back to its "Initialization Status upon Turn ON", -- EXCEPT for the TOP OF FORM setting. The total number of these CANCEL commands depend upon the total number of the Special Fonts and Print Styles that the program is to use. If you dont USE a Printer Function, then it isn't necessary to CANCEL that function. The three Print Styles (for this printer) does not need to be cancelled.

QL Users take note: The protocol for sending control codes to a printer is different. The PRINCIPLE though is the same. The printer must be hit with a CONTROL CODE to respond and change an internal setting. We do that with the TS-2068 (Aerco and Oliger Interface) by <OUT 127,code>. Some other TS-2068 interfaces use <LPRINT CHR$ code>. A printer, such as the KXP-1124 may be "an EPSON Compatable" printer, but that does not mean that its required Command Codes are the same as an Epson Printer. Most printers require different Control Codes to do similiar or identical internal changes. There are three methods for sending Control Codes to the Printer. QL users must use PRINT CHR$ number to the Serial Port. TS-2068 users of Aerco and Oliger Printer code sends these commands via <OUT 127,number>.

Other TS-2068 users of TASMAN or other prcode may need to use LPRINT CHR$ (number). If you use TASMAN or another interface than Aerco or Oliger, then consult your Interface manual for the proper way to send command characters to the printer. We will use the method for the TS-2068 Aerco or Oliger interfaces. The next is where your PRINTER MANUAL is the KEY. You must decide WHICH of the special fonts and styles that you want to have done automatically with the menu control, then go about using the printer manual to dig out both the "SET and CANCEL" comands. You can use the rationale given herein and corrolate with your printer manual to develop the programming changes in these line numbers "that will command your printer" to accomplish the style changes. But the following NEEDS are Common to all.

COMMANDING THE PRINTER

The first part of this Printer Control Section is to Set the Printer Back to its Turn On STATE, without resetting the TOP of FORM. Then printing can resume with a new style of print and the software can retain PAGE CONTROL. You can see by the Menu Illustration the styles and fonts that "I selected" for my Printer. Now the first task to do is to CANCEL all of these settings. At Line 8034 we start by <GO SUB BU> which throws you for a loop, eh? Well, "bu is line #8080", and lets



look at that.

The purpose of line #8080 is to check to see if the PRINTER IS BUSY. The printer goes through a routine of "turning itself off or on" to do internal changes. ONE Control Command sent from the keyboard is acted upon by the printer faster than one can type and send the next command. But, in software lines, the computer is sending control codes faster than the printer can react to. Therefore a LOOP is needed to check the Printer's status before sending the NEXT control command. The TS-2068 uses the IN command through "Port 127". Most TS-2068's will return "253" if the printer is BUSY or OFF LINE.

To check yours, first turn ON your Printer, then INITIALIZE your PRINTER CODE. Then leave your printer ON, but switch it OFF LINE. Then type: <PRINT IN 127 ENTER>. "IN" is the token below the I Key. If you get some other number than 253, then use that number instead of 253 in line 8080. Now we can go back to line #8034 to continue to CANCEL all printer functions that we will use in the "stymn.B6" program. The programming is used <OUT PO,18>. The variable "po=127", so that means <OUT 127,18>. This is to CANCEL the Compressed Mode of print. Between each OUT command, <GO SUB BU> is used to loop line 8080 until the printer is READY. So, we wont discuss the <GO SUB BU> any more. Next in line #8034 the grouped command <27,119,0> is sent to the printer in similiar fashion with <OUT PO>. This Cancels the "Double High character" Mode.

Next <27,53> is sent to CANCEL ITALICS Mode. Then <27,45,0> is sent to CANCEL UNDERLINE MODE. Now this is all of the special settings that is used in this program that are not "self cancelling when an alternate selection is made". If you use more in your programming then you must include CANCEL COMMANDS in this line of programming. AND- your printer may require all or some more of the print style settings to be CANCELLED before another can be set. An Okidata Printer, for example, requires the existing print style to be CANCELLED before another style can be set.

Line #8036 begins the "PRINTER SET Commands". The first group, <27,120,0> sets LETTER QUALITY Mode. Then <27,107,num> sets one of two font styles (Bold if in Pica or Elite) or COURIER if in Micron Pitch. Line 8038-8044 sets the print style selected at menu. Then the programming RETURNS to LINE 8024 where another GOSUB is to Line 8050. At line 8050-8056 the special fonts are set, compressed, double height, Italics, or Script. Then RETURN goes to line #8026 where the Matching Line length is calculated, and tab and page center is set. The Style Menu is again presented where it is appropriate to select a return to the Format Menu, as the work is done. On the way back to the Style Menu, line #8114 DELETES programming from memory. The Format Menu and the Style Menu work together as a suite for both Format and Printer Control. _BJ-

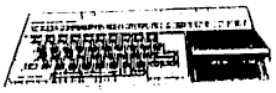
SUMMARY

The use of MERGE and DELETE together can broaden your programming horizons. By the use of these powerful functions, you can bring utilities aboard a running program, use them and then discard them. This lets your software have the most data capacity. Printer Control programming is needed only once or twice while running a program. Why clutter up memory by keeping un-needed programs on board? Pardon me for showing off with my new printer! This entire article was printed non-stop with this printer and the Daisy.B6 software. I hope that this discussion of Printers, their control, and Formatting helps. If you get anything from this I hope that it is "Dont be afraid of a new printer" and "You can make it work". _BJ-

Left Handed Ways of Working a Printer

This small group of cognizibles are to given to provolk thought. I Just got a new panasonic KXP-1124, 24 pin dot matrix printer. The first job was to modify my TS-2068 software for the new printer. That done, I tried the printer with the QL. It worked right out of the box with the Quill Config set-up. But there was no way of Setting the Printer's modes with the Quill Software. Next, I did some printing and became confused about how to set print pitches from the front panel. My set-up uses a Centronics "A/B" switch to switch between the QL and the TS-2068 to the printer. The bright idea came; "why not use my TS-2068 Daisy software to set-up special print pitches and Letter Quality mode, and then throw the the A/B switch over to the QL for printing?"
-So, here goes. Daisy booted in and the Style Menu was used to set Micron 120 Style, which gives 120 character lines (17 Pitch). The A/B switch was moved to the QL position, and this columm is the result.

Cont. on Pg 23



A HALF DOZEN OR MORE TOS TOOLS BY EARL DUNNINGTON

```
100 REM TOS TOOLS #1-5
102 REM REQUIRES OS-64 CART, CENTRONICS
INTERFACE & PRINTER
104 CLEAR 63423: LET A=0: LET B=0: LET
C=0: LET U$=""
9800 REM TOOL #1-PRINTER INITIALIZATION
ROUTINE
9801 GO SUB 9820: PRINT #0;"TURN OFF PRI
NTER"'"SET PRINTER SWITCHES"'"IF NECESSA
RY"'"THEN <ENTER>": PAUSE 0: CLS
9804 PRINT #0;"INPUT TYPE OF INTERFACE"'
"<1> AERCO OR OLIGER"'"<2> TASMAN B"'"<3
> TASMAN C"'"<4> A & J": PAUSE 0: LET ZZ
=VAL INKEY$: IF ZZ<1 OR ZZ>4 THEN  CLS :
 GO TO 9801
9806 POKE 65523,(167 AND ZZ=1)+(135 AND
ZZ=2)+(122 AND ZZ=3)+(185 AND ZZ=4)
9807 CLS : INPUT "LINE WIDTH?"'"THEN <EN
TER> ";LW
9808 INPUT "LEFT MARGIN TAB?"'"THEN <ENT
ER> ";TB: IF LW+TB>80 OR LW+TB<1 THEN  P
RINT #0;"LINE WIDTH + LEFT MARGIN TAB MU
ST BE LESS THAN 81 WITH PICA"'"<ENTER> T
O CONTINUE": PAUSE 0: GO TO 9807
9809 POKE 65525,LW+TB: GO SUB 9821: RAND
OMIZE USR 15623: REM The OS-64 does not
have a single poke for setting left marg
in. Must know the printer code if any
9810 GO SUB 9820: INPUT "<1> COMPUTER LI
NE FEED?"'"<2> NO LINE FEED"'"THEN <ENTE
R> ";LF: IF LF<1 OR LF>2 THEN  GO TO 981
0
9811 POKE 65510,(13 AND LF=1)+(0 AND LF=
2)
9812 INPUT "<1> COMPUTER CARRIAGE RETURN
?"'"<2> NO CARRIAGE RETURN"'"THEN <ENTER
> ";CR: IF CR<1 OR CR>2 THEN  GO TO 9812
9813 POKE 65511,(10 AND CR=1)+(0 AND CR=
2)
9814 PRINT #0;"TURN ON THE PRINTER, THEN
 <ENTER>": PAUSE 0: GO SUB 9821: LPRINT
: CLS
9815 FOR N=1 TO 5: IF TB=0 THEN  GO TO 9
818
9816 DIM M$(TB): LPRINT M$;
9818 FOR M=1 TO LW: LPRINT "X";: NEXT M:
 NEXT N: LPRINT : REM Test print lines f
or printer whose left margin code is not
 known
9819 STOP
9820 CLS : OUT 255,0: OUT 244,0: BORDER
0: PAPER 0: INK 5: CLS : RETURN : REM Pu
t OS-64 to sleep
9821 CLS : OUT 255,62: OUT 244,3: INK 7:
 CLS : RETURN : REM  wake up OS-64
```

This program was converted from Bill Jones "LKDOS TOOLS BY THE HALF DOZEN" published in the July issue. Whereever possible I have used the same range of line numbers.

There are many differences between LKDOS and TOS. Rather than using RANDOMIZE USR 100, TOS uses an asterisk after the BASIC command to denote that it is a disk command. TOS has a Serial Printer driver and does not have a Centronics Parallel print driver. This program is designed for use with the OS-64 cartridge, a Centronics Parallel printer and a 80 column monitor.

With the OS-64 in the dock, when the 2068 is powered up, the screen colors are set up with a 64 character display. The default printer interface is either an AERCO or Oliger. The two problems with the OS-64, is that one should have a 80 column monitor and that there is no single POKE for setting the left margin of the printer. One must know the manufacturers code for setting the left margin or program around it. In order to make this program universal as far as printers go, I have chosen the second option.

The first two lines in the program are REM statements. Line 104 sets RAMTOP as the OS-64 V1.71 does not do this automatically. This prevents unwanted crashes while in the BASIC program. Aswith LKDOS Tools, variables are initialized so they can be found faster by the program.

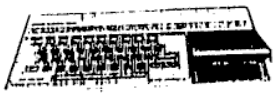
TOOL #1. PRINTER INITIALIZATION. TOS does not require initialization like LKDOS. Merely turning on the drives and inserting a formatted or program disk accomplishes this. The OS-64, however, must be initialized for type of interface, number of characters per line, if the computer must generate a carriage return, and if the computer must generate a line feed. This is accomplished in lines 9801 to 9814.

The next tool begins at 9815 ending at 9819. Itis merely a five line proof test of the initialization of the OS+64 and the printer. To overcome the lack of a single poke to set the left margin, the string variable M$ is dimensioned using the variable TB. When a string is dimensioned it contains only spaces. M$ is printed at the front of each line to provide the left margin.

The next tool is the two really neat subroutines, devised by Bill Jones, that allow one to switch from a 64 character display to 32 and vice versa. They are contained in lines 9820 and 9821. When in the 32 character mode, you cannot use LOAD, SAVE, LPRINT, or RANDOMIZE a location in the OS-64 without crashing the computer.

TOOL #2. The CAT print tool starts at line 9822. As the old saying goes, there is more than one way of skinning a cat. In order to overcome the margin setting problem, a different method is used from the one used by Bill in his LKDOS CAT Print.

This version picks up the characters from the screen using



```
9822 REM TOOL #2-CAT PRINT-Print a Disk
Catalog
9823 PRINT #0;"ARE THE DRIVES ON AND IS"
'"THERE A DISK IN ""A""DRIVE?"''TURN OFF
 PRINTER"''<ENTER> WHEN READY": PAUSE 0:
 CLS : REM TOS Initialization
9824 POKE 65525,80: RANDOMIZE USR 15623:
 PRINT #0;"TURN ON PRINTER THEN <ENTER>"
: PAUSE 0: CLS : REM Set LPRINT width to
 80
9826 PRINT #0;"INSERT DISK TO BE CATALOG
ED"''IN DRIVE ""A"" THEN <ENTER>": PAUSE
 0 : CLS : GO SUB 9856: GO TO *":": CAT
*: GO SUB 9860: GO TO *"A"d
9830 GO SUB 9856: CLS : CAT *: GO SUB 98
60: INPUT "IS THE DISK TREE STRUCTURED"'
"<1> YES"''<2> NO"''THEN <ENTER> ";T: IF
 T<1 OR T>2 THEN  GO TO 9830: REM Does t
he catalog show subdirectories?
9832 IF T=1 THEN  GO TO 9840
9834 INPUT "<1> CATALOG ANOTHER DISK?"''
<2> QUIT?"''THEN <ENTER>";S: IF S<1 OR S
>2 THEN  GO TO 9834
9836 IF S=1 THEN  GO TO 9826
9838 GO TO 9864
9840 INPUT "<1> LPRINT ANOTHER CATALOG?"
'"<2> QUIT"''THEN <ENTER>";S: IF S<1 OR
S>2 THEN  GO TO 9840
9842 IF S=2 THEN  GO TO 9864
9844 GO SUB 9856: INPUT "INPUT SUBDIRECT
ORY PATHNAME"''THEN <ENTER> ";P$: CAT #P
$: GO SUB 9860: GO TO 9840
9856 INPUT "LPRINT"''<1> LEFT SIDE"''<2>
 RIGHT SIDE"''THEN <ENTER>";S: CLS : IF
S<1 OR S>2 THEN  GO TO 9856
9858 RETURN
9860 DIM M$(5): IF S=2 THEN  DIM M$(43):
 REM Sets margin
9862 DIM C$(32): FOR N=0 TO 21: FOR M=0
TO 31: LET C$(M+1)=SCREEN$ (N,M): NEXT M
: LPRINT M$;C$: IF C$(1 TO 5)="Max 1" TH
EN  RETURN : REM Picks up catalog from s
creen and prints it
9863 NEXT N: RETURN
9864 STOP

9868 REM TOOL #3-SENDING STRING DATA TO
DISK
9869 REM Create the U$ Data first " Use
CAPS SHIFT+SYMBOL SHIFT to QUIT " " Use
ENTER for starting new line "
9870 POKE 23730,191: POKE 23731,247: LET
 A=0: LET B=0: LET C=0: CLS : INPUT "TUR
N DRIVES ON, INSERT DATA DISK"''ENTER A
FILE NBR <1 TO 99999>";ZZ: GO SUB 9970:
LET U$="": CLS : PRINT AT 21,60;LEN U$;A
T A,B;"_";: REM TOS allows 8 character n
ame + period and 3 character file type e
xtension
```

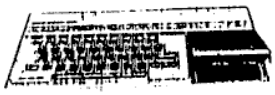
SCREEN$. In addition Tree Structured or nested Subdirectories are allowed in TOS. Not so in LKDOS. Unfortunately Tree Structured directories are stressed in the TOS manual, rather than just formatting a disk, SAVEing, and LOADing a program. This makes the use of disk drives appear very complicated to the novice. Actually using disk drive "A" is no more complicated than using a tape recorder to save and load programs. The drives in TOS are daisy chained to the "A" drive. That is a program disk must be in the "A" drive for the other drives to work. The directory in another drive becomes a subdirectory of the disk in the "A" drive. Printing a catalog from any other drive would give an erroneous result, listing a directory that is not on the disk. For this reason I did not allow the option of a change in drives.

TOOL #3-SENDING STRING DATA TO DISK. This program follows the LKDOS version fairly closely. The text entry display is changed to provide a 64 character display. Line 9902 inserts an ENTER character into U$ as an end of record marker. TOS allows 65535 records to a data file. TOS uses a disk drive buffer that allows only 256 characters to be sent to the disk at one time. Lines 9903 to 9904 utilize string slicing and a loop to send the string data to the disk. Line 9940 contains the 2068 replacement for the TS 1000/1500 SCROLL command (LET t=USR 2361). When the text input fills the screen to the end of line 20, the screen scrolls upone line, leaving the character counter undisturbed. Line 9971 concatenates (chains the parts) of the file name into N$ and opens the file as a sequential streem file (no fixed record length) in the output to disk mode. TOS allows an eight character name plus an extension consisting of a period and a three character data type. The use of the extension with BASIC program names is optional.

TOOL #4-LPRINT OR VIEW A DISK FILE. In the LKDOS version Tools 3 and 4 are intertwined. As I wished to make a comparison of the two systems easy, I left them that way in the TOS version. I found it convenient, while programing, to be able to view or LPRINT any data file. Also to see the Catalog while choosing a file. I added these features in line 9920. Once again as the TOS buffer accepts only 256 characters at one time and due to the OS-64 laft margin problem, the program differs from the LKDOS version. TOS allows the use of 16 channels. Four fast (#1-#4) and 12 slow (#5-#16). The fast channels are more convenient for reading back (Input) a stream sequential file and so I used Channel #1. As the BASIC system does not recognize the TOS generated error reports, ON ERROR GOTO is not used. To determine when the the End of File is reached, lines 9930 and 9934 illustrate the method used for TOS.

TOOL #5. It is merely a subroutine for quick printer format changes and is contained in line 9960. Once again it differs from the LKDOS version because of that left margin problem of th e OS-64 and the Smith Corona Fastex 80 dot matrix printer.

This article was typed into data files using Tool #3 and Lprinted using Tool #4. One should remember that the major part of programing is the development of the logic which was done by Bill Jones in his LKDOS version. Many thanks Bill.



```
9872 POKE 23611,220
9874 IF PEEK 23611<221 THEN  GO TO 9874
9876 LET C=PEEK 23560: POKE 23611,220
9878 IF C=12 THEN  GO TO 9890: REM delet
e char
9880 IF C=13  THEN  GO SUB 9940: GO SUB
9902: GO TO 9872: REM ENTER key
9882 IF C=14 THEN  GO SUB 9902: GO TO 99
06: REM CS+SS keys
9884 IF C<32 OR C>122 THEN  GO TO 9872
9885 PRINT AT A,B;CHR$ C;: LET B=B+1: IF
 B=64 THEN  GO SUB 9940
9886 PRINT AT A,B;"_";: LET U$=U$+CHR$ C
: PRINT AT 21,60;LEN U$
9888 GO TO 9872
9890 IF LEN U$=0 THEN  GO TO 9872: REM e
rror trap
9891 LET U$=U$( TO LEN U$-1): PRINT AT 2
1,60;"    ";AT 21,60;LEN U$;AT A,B;" ":
LET B=B-1: IF B=-1 THEN  LET B=63: LET A
=A-1: IF A=-1 THEN  LET A=0
9892 PRINT AT A,B;"_": GO TO 9872
9902 LET U$=U$+CHR$ 13: REM End of recor
d marker
9903 FOR N=1 TO LEN U$ STEP 256: IF LEN
U$<N+256 THEN  PRINT #5;U$(N TO LEN U$)
: GO TO 9905: REM TOS Drive buffer accep
ts only up to 256 characters at one time
9904 PRINT #5;U$(N TO N+255): NEXT N
9905 LET U$="": PRINT AT 21,60;"    ": R
ETURN
9906 CLS : INPUT "<1> CREATE MORE TEXT"'
"<2> QUIT";XX: IF XX=1 THEN  GO SUB 9940
: GO TO 9872
9908 CLOSE #5: STOP
9918 REM TOOL #4. LPRINT OR VIEW A DISK
FILE
9920 CLS : CAT #: INPUT "ENTER THE FILE
NBR <1 TO 99999> ";ZZ: INPUT "<1> VIEW F
ILE <2> LPRINT FILE ";XX: LET U$="": IF
XX=1 THEN  GO TO 9928
9924 CLS : PRINT #0;"TURN PRINTER ON, TH
EN <ENTER>": PAUSE 0: GO SUB 9960: REM
printer initialization
9928 CLS : LET U$="": GO SUB 9980: REM O
pen disk file for input to screen or pri
nter
9929 IF XX=2 THEN  GO TO 9933: REM  send
 data to printer instead of screen
9930 POKE 23729,255: INPUT #1;U$: IF PE
EK 23728<>0 THEN  GO TO 9946: REM Peek a
nd Poke are error trap as ON ERROR does
not recognize TOS system errors
9931 PRINT U$;: GO TO 9930
9933 IF LM>0 THEN  LPRINT M$;
9934 FOR N=1 TO LW: POKE 23729,255: INPU
T #1;U$: IF PEEK 23728<>0 THEN  LPRINT
: GO TO 9946
9935 LPRINT U$;: IF U$=CHR$ 13 THEN  GO
TO 9933
9936 NEXT N: GO TO 9933
9940 LET A=A+1: LET B=0: IF A>20 THEN  P
RINT AT 21,60;"    ": LET t=USR 2361: LE
T A=20: PRINT AT A,B;"_": RETURN
9942 PRINT AT A,B;"_";: RETURN
9946 CLOSE #1: PRINT #0;"REACHED END OF
 FILE, PRESS <ENTER>": PAUSE 0
9950 STOP
9952 GO TO 9920
9956 REM TOOL #5. GOSUB ROUTINE FOR QUIC
K FORMAT CHANGES
9960 CLS : INPUT "INPUT TAB ";LM: INPUT
"LINE WIDTH ";LW: POKE 65525,LM+LW: RAND
OMIZE USR 15623: IF LM>0 THEN  DIM M$(LM
)
9962 LPRINT : RETURN
9970 REM Assign file name (N$) and open
TOS file as a Stream File
9971 LET N$="DTA"+STR$ ZZ+".DAT": DIM #N
$: OPEN #5;N$;0: RETURN
9980 REM Assign file name N$ and open th
at TOS file for (I)nput of (1) character
 at a time, to screen or printer using c
hannel one (#1)
9981 LET N$="DTA"+STR$ ZZ+".DAT": OPEN #
#1;N$;I;1: RETURN
```

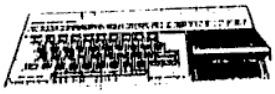

And Thanks TO you Earl -

PRINTERS (FROM PAGE 20)

Speaking of Printers and their manipulation: My next project is to use a 64K Printer Buffer and put it IN LINE with an A/B switch. Then I should be able to feed data to the buffer from either computer. Try this idea: Create two documents, one in the QL and another in the TS-2068. Send the QL doc to the Printer buffer. Then send the TS-2068 doc to the printer buffer. The two docs stack up in the buffer and should print as one longer document.

Now that may be like be like riding two bicycles at the same time, eh? Well, what about this- We who have two computers often have documents created with Quill or our TS-2068 software. The doc files are in two floppies and are not transferrable between the two computers. Wouldn't it be nice to "append a QL Quill file" with a TS-2068 file in disk? Could save a lot of re-typing or paste up.

(Durned if it didn't work!)



GRAPH of a POLYNOMIAL

Polynomials are power equations of the type most often written in the form of.....

aX↑b + cX↑d + eX↑f + .. +g = 0

where a, c, e, and f are usually small whole, positive or negative numbers, and b, d, f, etc are interger exponents arranged in descending order of power.

Most often some one value, or several values, of X satisfies this relationship. Which is to say that such value of X, when substituted into the polynomial makes its overall value equal to the zero on the right hand side of the equal mark. These values of X are known as roots of the equation, or simply roots, and they can be rational or irrational numbers. Irrational numbers, you may recall, are those which cannot be expressed precisely. Some well-known examples are....

PI = 3.1415927.....
e = 2.7182818.....

If equated to Y instead of 0, equations of this type can be plotted on coordinated X- and Y-axes paper, but the method is usually tedious for any equation over the second power, one where the first exponent, b, is greater than two. One of the purposes of plotting such an equation is to determine the points where the curve crosses, or touches, the X-axis. Such points, where the value of X makes Y = 0, are roots of the equation.

There may be as many real roots to a polynomial as represented by the power of the first term in X. Thus an equation of the second power can have two real roots, but it may also have none. An equation of the third power may have three real roots or one. It will not have less. An equation of the fourth power may have four, two, or no real roots. All of this depends upon the value of the last term of the equation. Its value can raise or lower the entire curve with respect to the coordinate axes,providing the other terms remain unchanged.

Algebra offers several methods of solving these equations. A few can be factored, and the like, but many of them can be solved (roughly) only by plotting and tedious trial and error calculations. Here is where the Timex shines. It can crunch numbers and speed them thru numerous and repetitious calculations. We will start by having the computer plot the significant part of the curve for us.

Program G-11 was devised to plot the curve of any power equation. Normally the TS2068 plots using pixels as though the lower left corner of the screen is the origin, the Y-axis the left edge, and the X-axis the bottom edge. Thus the normal Timex screen is the first quadrant. Hence we must first center the axes. In our case we arbitrarily moved the X-axis up by 86 pixels and the Y-axis to the right by 130 pixels. The values mentioned are incorporated into lines 20 and 140 of the program G-11.

Next there is a matter of scale. Power equations can develop very large positive or negative values of Y for relatively small values of X. For this reason vertical and horizontal scale factors are required. These are entered into the program in lines 100 and 110, as V and H. The corresponding scale markings on the graph appear in lines 70 and 80. The relationship between factor and scale marking is.....

V - Factor	Y Scale Mark
.2	200
.4	100
.8	50
1.0	40
2.0	20
4.0	10
8.0	5
20.0	2

H - Factor	X Scale Mark
5.0	20
10.0	10
20.0	5

For starters, use the combination of factor and scale mark, starting with the top-most pair. If the resulting curve is too pinched or compressed, select a combination further down the table. Line 135 prevents those annoying out-of-range error reports.

Note that the program lines subject to change with each new equation have been emphasized by having their command words in inverse lettering. Do not enter a new line by eradicating the old one. Bring the old one down and change it, leaving the inverse lettering intact.

Figure 1 shows the basic program, G-11, built around the equation.....

Y = 16X↑3 + 16X↑2 + 3X

Notice that its curve shows three real roots. If the equation had an additional constant term of say +1, the whole curve would move up with respect to the axes and it would have only one real root.

Program G-11A, Figure 2, shows the graph of a different equation.....

Y = 2X↑3 + 11X↑2 -31X - 180

This one has three real roots which happen to be X = -5, -4.5, and +4. They don't usually come out in such nice little, rational numbers. The first two roots become apparent when the area from X = -6 to X = -4 is expanded by using a vertical factor of V = 4. See Figure 4, for a blow-up of this area, so treated. Figure 3 makes the curve look tangent to the axis. Its scale is too small.

G-11B, Figure 3, shows the same basic program with a fourth power equation.....

Y = 6X↑4 + 5X↑3 -5X↑2 - 5X -1

The graph of this curve shows that it has four real roots. If the last term is changed from -1 to +3, all of the four real roots disappear. Make this substitution and observe the resulting curve.

Here are a few more polynomials of this variety to play around with.....

Y = X↑2 - X - 5
Y = X↑4 - 13X↑2 + 35
Y = X↑3 - 4X↑2 - 2X + 7
Y = X↑3 - 4X↑2 - 2X - 3

Save this paper for future reference. Programs to evaluate these roots, with a high degree of precision, will be the subject of a future tutorial.

Warren Fricke

```
  5 REM ** "G-11", 7-18-89, WF
 10 REM ** GRAPH of a POWER
           EQUATION
 20 PLOT 0,86: DRAW 0,-86: DRAW
 255,0: DRAW 0,86: DRAW -255,0:
DRAW 0,89: DRAW 255,0: DRAW 0,-8
9: PLOT 130,0: DRAW 0,175
 40 FOR n=26 TO 146 STEP 20: PL
OT 129,n: PLOT 131,n: NEXT n
 50 FOR n=30 TO 230 STEP 20: PL
OT n,87: PLOT n,85: NEXT n
 60 PRINT AT 1,1;"GRAPH of..
 Y";AT 12,30;"X"
 70 PRINT AT 5,17;"+2";AT 16,17
;"-2"
 80 PRINT AT 9,2;"-5";AT 10,28;
"+5"
 100 LET V=20
 110 LET H=20
 120 FOR x=-1 TO 1 STEP .01
 130 LET y=16*x*x*x+15*x*x+3*x
 135 IF ABS (y*V)>85 THEN GO TO
150
 140 PLOT x*H+130,y*V+86
 150 NEXT x
 200 PRINT AT 3,1;"Y = 16X↑3 + 1
6X↑2 +3X"
```

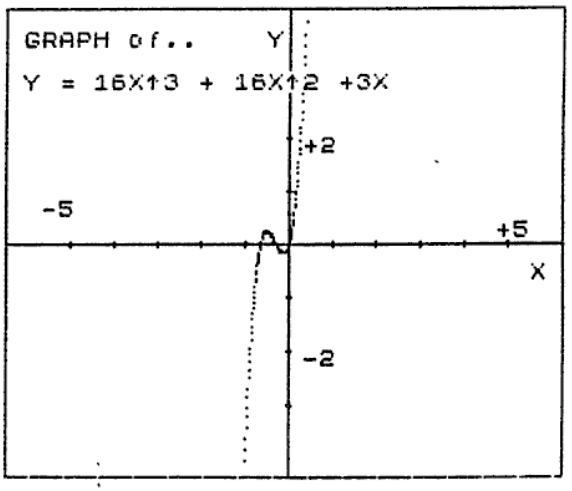


FIGURE 1 Cont. Next Page

Musings

Now you know ole BJ has a penchant to ramble, eh? Well there is some scattered space to fill, so this will be continued all over the magazine. I'll try to make the epistles worth the space. First, I want to apologize to several writers whose articles did not appear in this issue. Be assurred that they WILL make it. Update is blessed with excellent article support by outstanding writers. I appreciate your contributions and I know that the subscribers do also. So, KEEP THEM COMING. Your articles WILL be published.

I was talking with Larry Kenny T'other late night and ask him a question. Larry came back with, "Its in your own magazine", and "dont you read Update?". (Arrugh!). This issue of Update gives me great problems. Bill Cable came in with over 20 pages of articles about the QL, and ALL about things that I dont know about! So, INDEED I'll have to read and study Update! Al Feng and his Networking will take some time to digest, but most of all it will take "DOING AT IT". And there is ole Warren Fricke trying to teach me polynomials! I think that I "used to know that" sometime in the distant past. Gotta study!

CONTINUED, PAGE 12

```
  5 REM ** "G-11A", 7-18-89, WF
 10 REM ** GRAPH of a POWER
           EQUATION
 20 PLOT 0,86: DRAW 0,-86: DRAW
 255,0: DRAW 0,86: DRAW -255,0:
DRAW 0,89: DRAW 255,0: DRAW 0,-8
9: PLOT 130,0: DRAW 0,175
 40 FOR n=26 TO 146 STEP 20: PL
OT 129,n: PLOT 131,n: NEXT n
 50 FOR n=30 TO 230 STEP 20: PL
OT n,87: PLOT n,85: NEXT n
 60 PRINT AT 1,1;"GRAPH of..
 Y";AT 12,30;"X"
 70 [illegible]6,17;"+100";AT 16,
17;"-100": REM ** Y-SCALE
 80 [illegible]9,2;"-10";AT 10,28
;"+10": REM ** X-SCALE
 100 [illegible]V=.4
 110 [illegible]H=10
 120 [illegible]x=-10 TO 10 STEP .1
 130 [illegible]y=2*x*x*x+11*x*x-31*x-1
80
 135 IF ABS (y*V)>85 THEN GO TO
150
 140 PLOT x*H+130,y*V+86
 150 NEXT x
 200 [illegible]3,1;"Y = 2X↑3 + 11
X↑2 - 31X - 180"
```

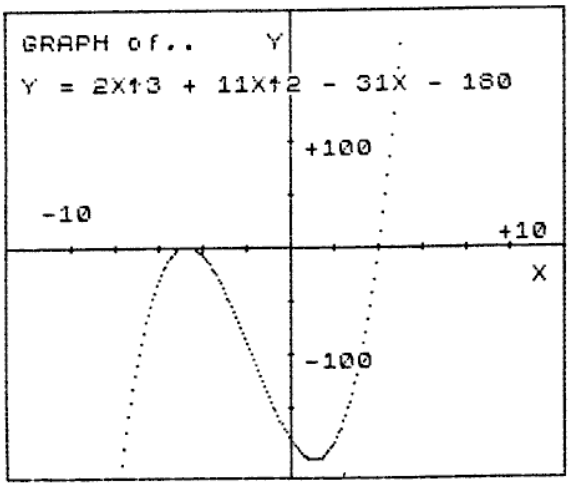


FIGURE 2

```
  5 REM ** "G-11B", 7-18-89, WF
 10 REM ** GRAPH of a POWER
           EQUATION
 20 PLOT 0,86: DRAW 0,-86: DRAW
 255,0: DRAW 0,86: DRAW -255,0:
DRAW 0,89: DRAW 255,0: DRAW 0,-8
9: PLOT 130,0: DRAW 0,175
 40 FOR n=26 TO 146 STEP 20: PL
OT 129,n: PLOT 131,n: NEXT n
 50 FOR n=30 TO 230 STEP 20: PL
OT n,87: PLOT n,85: NEXT n
 60 PRINT AT 1,1;"GRAPH of..
 Y";AT 12,30;"X"
 70 [illegible]6,17;"+2";AT 16,17
;"-2": REM ** Y-SCALE
 80 [illegible]9,2;"-5";AT 10,28;
"+5": REM ** X-SCALE
 100 [illegible]V=20
 110 [illegible]H=20
 120 [illegible]x=-2 TO 2 STEP .01
 130 [illegible]y=6*x*x*x*x+5*x*x*x-5*x
*x-5*x-1
 135 IF ABS (y*V)>85 THEN GO TO
150
 140 PLOT x*H+130,y*V+86
 150 NEXT x
 200 [illegible]3,1;"Y = 6X↑4 + 5X
↑3 - 5X↑2 - 5X -1"
```

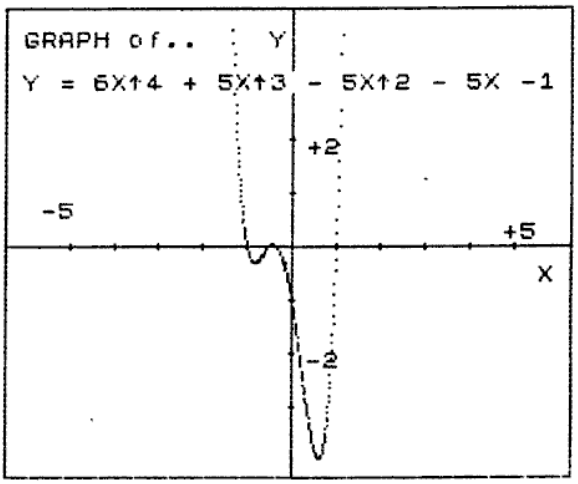


FIGURE 3

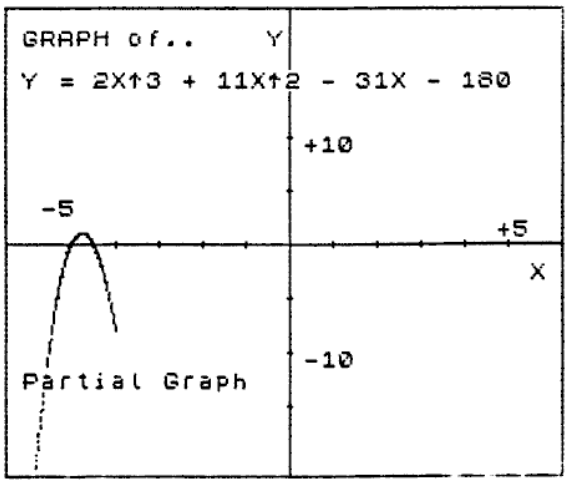


FIGURE 4

FOR SALE- QL and TS-2068 Items

TS-2068 TOS DISK DRIVES, AMDEX DRIVES, TOS CONTROLLERS AND POWER SUPPLIES, 3" Diskettes. SINCLAIR QL COMPUTER and other Misc. TS-2068 and QL Items. For list and price, send SASE or Call: Stan Light, PO Box 13, Ellington, FL 34222. Tel: 813-747-5833.

WOOD AND WIND COMPUTING

Bill Cable : RR 2 Box 92 : Cornish, NH 03745 : 603 675 2218

CABLE COLUMN

This issue of UPDATE initiates me as the QL Editor. I am happy to assist Bill Jones in producing this magazine so important to North American QL users. Late summer is a very busy time for me. Have to prepare for winter. I hope to be able to put more time into the job in the next issue. It is important that we get maximum participation from the QL base we have. We have QL users at all levels of expertise and with many different configurations of equipment. As editor I want to provide something for everyone and a lot for most. My background is mostly with computer software and even then there is a lot I don't know. I hope I will get help, advice, criticism from readers. I would like programs to print for readers to type in. I would also like to print reviews of hardware and software. And tips about perks and quirks of the QL. Also I need to know what readers want to see. One quality of UPDATE that I like is its friendly non pretentious flavor. We all have a lot to learn about using our QLs more effectively and UPDATE is a forum for doing it.

The first computer I ever owned was a QL that I bought in 1986. So I am relatively new to personal computing and Sinclair computers. I have a degree in mathematics (1967, University of Florida). That was before computer courses were offered on campus. But my various jobs led me into computer programming. Mostly in FORTRAN on DEC machines but also in BASIC at Dartmouth College (the birthplace of BASIC). I got burned out programming for others by 1980 and started a cabinetry shop. But I wanted to get back into computing for myself on a part time basis and started looking around for a computer. Having limited resources I was attracted to the QL. It was selling for $150 when I bought my first one. I was quite impressed. I felt like I had almost as much power as I did on the large computers I used to work on and a much nicer language and operating system to work with. The PSION programs that came with it were very good also. Being so close to A+ (source of all US QLs) I became a dealer and sold QL's locally and worked with several businesses writing custom ARCHIVE programs for them. Then I started selling my software nationally on my own, through Peter Hale's EMSOFT, and finally as issue disks for UPDATE.

Writing software takes an enormous investment in time up front. You have to have it before you can sell it. Writing major pieces of software for the North American QL market stands little chance of much financial success and so has to be done during spare time. Active QL users probably number less than 2000. That is a pretty small number to write software for. I would like to make a request to all QL users for all of us writing software for North America (Peter Hale, Bob Hartung, Al Feng, David McCall, myself, and others). **Please consider our software.** Get a catalog from EMSOFT and check out the advertisements and issue disks descriptions in UPDATE. See what we have to offer. We are trying to provide a service to QL users. We have the ability to write software as good as any coming from overseas. We all do this on the side but could put more energy into it if we had more financial success. You have one great advantage in dealing with us. We can be contacted by mail or phone about our software and can respond to your specific problems. We not only sell programs but are interested in helping all QL owners get the most from their computers. And we are definitely doing it out of love for the QL and not for the money.

BOOTING UP : Programs to help

In the description below I will use 'mdv1_' in the description. Those with floppy drives can substitute 'flp1_'. Each time we turn on our QL or reset it

we have a chance to boot up. Load a program from mdv1_ called 'boot' which gets things going. So a boot program is a SUPERBASIC program named 'boot' that automatically starts a more complex (usually executable machine code) program(s). All we have to do is press <F1> or type 'lrun mdv1_boot<ENTER>. The PSION programs (QUILL, ARCHIVE, ABACUS, EASEL) have boot programs like this. The real PSION machine code programs are called qlwp, qldb, qlss, qlbg and each cartridge has a boot program which starts execution of the machine code. Often it would be more convenient if we did a few things : set time/date, activate a toolkit, load speedscreen, etc before starting the program. Everyone should make sure the time/date is correct at boot up. It is irritating to start a program that makes use of the time/date only to find that it is wrong and we can't set it from within the program. Toolkits and disk interfaces often time/date stamp files when saved. This can sometimes be very important in identifying a file. The TRUMPCARD/TK2 Toolkit has print buffering, 'hot key' screen dump, and 'hot key' macro capability plus much more. We should set these features on boot up. Many of you may wonder what are these toolkits everyone talks about. They are extensions to SUPERBASIC and QDOS that give new useful commands or make corrections to ROM bugs. They may be on ROM or in the disk interface, or loaded into RAM at start up. It usually takes some action on our part to activate them. Usually the best time to do that is at boot up time. Depending what program(s) we will be running we will want various toolkit features initialized. It usually makes sense to reboot upon quitting a program unless multitasking or saving files in RAM disk. The system is 'clean' after a reboot.

The programs described below should be helpful to all configurations of QLs. You incorporate them on the microdrive of the program you want to boot up on. Each of the PSION microdrives for instance. The current boot program on your microdrive must be renamed with a name of your choice (or copied to new name then 'boot' deleted). The 3 programs described below then need to be copied over. They will help you easily set the time/date and activate your toolkit if your have one. In this instance I used some of the features of my TRUMP CARD/TK2 system and microdrive version of speedscreen to illustrate a QUILL application. If you have a different toolkit, no toolkit, no cartridge speedscreen, or a different program to set up for you can adapt the initializing portion of the boot program (lines 1170 to 1390) to your needs.

I tried to make a boot up system that can be easily adapted to each boot up situation. It consists of 3 programs plus your own boot program which is renamed. They are :

1.	boot	initializes tool kit, verifies date, merges last_date and date_set if needed, starts old boot program which has been renamed.
2.	boot_date_set	prompts, sets, saves time/date
3.	boot_last_date	stores last time/date setting
4.	(your old boot)	starts up your program as it use to. Just renamed. On your QUILL cartridge (disk) you could rename boot to boot_quill.

This boot system has the following features :

1. Quick loading. That is why boot_date_set is only merged if needed.
2. Single key presses for quick entry.
2. Initialize (activate toolkit features, speedscreen)only if desired.
3. Force time/date set if out of range (as when first powered up).
4. Verify time/date if within range (as after a reset).

5. Use standard time (am/pm) instead of military time in displays.
6. Loads and runs a preset program (your original boot program).

From SUPERBASIC type in the programs as listed. Save each one when finished by its name. The line numbers of boot and boot_date_set should not overlap because they are sometimes merged. You need to have a full screen for window #2 for program entry. **'window #2,512,206,0,0 : cls #2<ENTER>'** will take care of that. If you don't have a toolkit the best way to start is **'auto 1000<ENTER>'**. With TK2 typing **'ed 1000<ENTER>'** gets you going.

After the 3 programs are typed in and saved you can pick a program to use it on. QUILL is what this particular code is set up for. Take your copy of QUILL (not the original) and put it in mdv1_. Have the cartridge of the boot programs you just entered in mdv2_. Now :

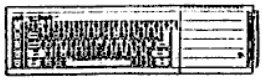
```
copy mdv1_boot to mdv1_boot_quill<ENTER>          (rename boot)
delete mdv1_boot<ENTER>                           (delete old copy)
copy mdv2_boot to mdv1_boot<ENTER>
copy mdv2_boot_date_set to mdv1_boot_date_set<ENTER>
copy mdv2_boot_last_date to mdv1_boot_last_date<ENTER>
```

Now you are all set. Pull the cartidges out reset. Put the modified QUILL in mdv1_ and press <F1>. If you find there is not enough room on the cartridge you can delete CLONE, INSTALL_BAS, CONFIG_BAS on your PSION copy.

If you are using a different program besides QUILL then substitute its name for QUILL and changes line 1040 of boot to reflect that change.

BOOT **program 1**

```
1000 REMark                        BOOT
1010 REMark  boot program : initialize tool kit, check/set date, start program      B Cable      9/89
1020 :
1030 LET boot_dev$ = "mdv1_"       : REMark boot device
1040 LET to_run$ = "boot_quill"    : REMark program to run after initializing
1050 LET sys_id$="MY SYSTEM"       : REMark your identification for the screen
1060 :
1070 LET year = 1989 : LET month = 12 : LET day = 20 : REMark  default time/date
1080 LET hour = 16 : LET minute = 30 : LET second = 0
1090 :
1100 LET l=7.889221E8 : REMark 1986 - The lower limit for good date
1110 LET u=1.104368E9 : REMark 1999 - The upper limit for good date
1120 :
1130 OPEN #1,con_512x206a0x0 : OPEN #2,con_512x206a0x0 : CLS #0
1140                       REMark querry for initialization
1150 head sys_id$ : CSIZE 0,0 : AT 2,55 : PRINT "for : ";to_run$
1160 AT 3,55 : PRINT "On  : ";boot_dev$ : CSIZE 2,1
1170 AT 7,0: CLS 4: PRINT "Initialize [y/n] : "; : PAPER 7 : INK 0 : PRINT " ";
1180 LET a$ = INKEY$(-1) : PRINT a$!;  : PAPER 0 : INK 7 : CLS 3 : CSIZE 0,0
1190 IF a$=="y"
1200                                  REMark  tk2_ext initialization
1210   AT  10,31 : PRINT "-Initialization-"
1220   AT 13,31 : PRINT "Activating TK2_EXT" : TK2_EXT
1230                                  REMark  screen dump with <ALT>d  key press
1240   AT 14,26 : PRINT "Setting up <ALT>d screen dump"
1250    SDP_SET 1,1,0,0: SDP_KEY d
1260                                   REMark  print buffer
1270   AT 15,29 : PRINT "Setting up print buffer" : PRT_USE ser,ser
1280                                  REMark  some altkey (hot keys for quill)
```



```
1290    AT 16,10
1300    PRINT "QUILL HOT KEYS : <ALT>p - prints, <ALT>t - top, <ALT>b - bottom"
1310    ALTKEY 'p',CHR$(240)&CHR$(112)&CHR$(10)&CHR$(10)&CHR$(10)
1320    ALTKEY 't',CHR$(240)&"G"&"T"
1330    ALTKEY 'b',CHR$(240)&'G'&'B'
1340    PAUSE 200
1350                                    REMark  Speedscreen initialization
1360    AT 17,31 : PRINT "loading speedscreen ..."
1370    LET x = RESPR(11240)
1380    LBYTES boot_dev$&"speedscreen_code",x
1390    CALL x
1400   END IF
1410                                  REMark if date within limits verify
1420 LET a$ = "y" :
1430 IF DATE>1 AND DATE<u
1440    head "VERIFYING TIME AND DATE"
1450    LET d$=DATE$ : LET d = d$(13 TO 14) : LET p$ = " am"
1460    IF d>11 : LET p$ = " pm" : IF d>12 : LET d = d-12 : END IF : END IF
1470    AT 4,5 : PRINT d$(6 TO 11)!!d$(1 TO 4);!!!DAY$!!!d;d$(15 TO);p$
1480    PRINT \\\"Change time/date [y/n] : "; : PAPER 7 : INK 0 : PRINT " ";
1490    LET a$ = INKEY$(-1) : PRINT a$!;   : PAPER 0 : INK 7 : CLS 3 : CSIZE 0,0
1500   END IF
1510                        REMark  merge last date, time/date setter if needed
1520 IF a$ = "y"
1530    AT 17,1: PRINT "Getting last date. If you get a 'not found' error type :"
1540    AT 18,1 : PRINT TO 25;"lrun ";boot_dev$;"boot_date_set<ENTER>"
1550    MERGE boot_dev$&"boot_last_date" : last_date : REMark get last date
1560    AT 17,1 : CLS 4 : CLS 2 : PRINT "merging "&boot_dev$&"boot_date_set ..."
1570    MERGE boot_dev$&"boot_date_set" : REMark merge date setting procedures
1580    date_set
1590  END IF
1600                                   REMark start program if any
1610    head "STARTING A PROGRAM" : CSIZE 0,0
1620    AT 10,1: PRINT "ENTER device_program to start [<ENTER> = ";to_run$;"] :";
1630    INPUT i$ : IF i$ = "" : LET i$ = boot_dev$&to_run$ : END IF : CLS : INK 5
1640    AT 6,1 : IF i$ = "" : PRINT "Done" : ELSE : PRINT "lrun ";i$ : END IF
1650    IF i$<>"" : LRUN i$ : END IF : STOP
1660 :
1670 DEFine PROCedure head(h$)
1680    CLS #1 : CSIZE 2,1 : PAPER 7 : INK 0 : AT 0,0 : CLS 4
1690    AT 0,20-LEN(h$)/2 : PRINT h$ : PAPER 0 : INK 7
1700   END DEFine head$
```

BOOT_DATE_SET **program 2**

```
5000 REMark                          BOOT_DATE_SET                    B Cable      9/89
5010 REMark Time/date setting procedures to be merged into boot program or
5020 REMark  will run as a stand alone program
5030 :
5040 LET boot_dev$ = "mdv1_"     : REMark  The boot device
5050 LET year = 1989 : LET month = 9 : LET day = 20
5060 LET hour = 18 : LET minute = 30 : LET second = 0
5070 :
5080 OPEN #1,con_512x206a0x0 : OPEN #2,con_512x206a0x0
5090 :
5100 date_set : CLS
5110 :
5120                        REMark prompt for all time/date fields
5130 DEFine PROCedure date_set
5140    LET what = 0 : LET second = 0 : CLS #0 : CLS : CSIZE 0,0
5150    AT 15,20 : PRINT " or ? single value, < or = move to min and max"
```

```
5160    AT 16,20 : PRINT "<space bar>  toggles to next time/date field"
5170    AT 17,20 : PRINT "1 or 2 digit number also works with auto toggle"
5180    AT 18,20 : PRINT "<ENTER>  when time and date are correct"
5190    AT 6,15 : PRINT "Date";TO 40;"Day";TO 59;"Time"
5200    AT 0,20 : CSIZE 2,1 : PRINT "SETTING DATE AND TIME"
5210    REPeat prompt_loop
5220      setter minute,31,0,59
5230      setter hour,28,0,23
5240      setter day,14,1,31
5250      setter month,10,1,12
5260      setter year,7,1962,1999
5270      setter second,34,0,59
5280      IF what = 10 : EXIT prompt_loop : END IF
5290    END REPeat prompt_loop
5300   CSIZE 0,0 : date_save
5310  END DEFine date_set
5320 :
5330                              REMark actual time/date prompter
5340 DEFine PROCedure setter(value,col,min,max)
5350   IF what = 10 : RETurn : END IF       : REMark <RETURN> key means no more
5360   REPeat setter_loop
5370     SDATE year,month,day,hour,minute,second
5380     display_date : CSIZE 2,1 : AT 5,col : PRINT ""
5390     LET what = CODE(INKEY$(-1))
5400     IF what=208: LET value=value+1: END IF          : REMark up arrow
5410     IF what=216: LET value=value-1: END IF          : REMark down arrow
5420     IF what=192: LET value=min: END IF              : REMark left arrow
5430     IF what=200: LET value=max: END IF              : REMark right arrow
5440                                  REMark see if numeric entry
5450     IF what>47 AND what<58
5460       LET what=what-48
5470       LET what2=CODE(INKEY$(100))  : REMark wait a little for 2nd digit
5480       IF what2 = 0
5490         LET value=what                        : REMark no 2nd digit
5500         IF max = 23 AND p$ = " pm" : LET value = value + 12 : END IF
5510         IF min = 1962 : LET value = value + 1990 : END IF
5520         LET what = 32                 : REMark advance to next time/date field
5530        ELSE
5540         IF what2>47 AND what2<58    : REMark check for 2nd digit
5550           LET what2 = what2 - 48
5560           LET value = 10*what + what2
5570           IF max = 23 AND p$ = " pm" : LET value = value + 12 : END IF
5580           IF min = 1962 : LET value = 1900 + value : END IF
5590           LET what = 32               : REMark advance to next time/date field
5600          END IF
5610        END IF
5620      END IF
5630     IF value>max: LET value=max: END IF     : REMark check if too large
5640     IF value<min: LET value=min: END IF     : REMark check if too small
5650                                             : REMark <space> or <RETURN>
5660     IF what = 10 OR what = 32
5670       AT 5,col : PRINT " " : RETurn
5680      END IF
5690    END REPeat setter_loop
5700  END DEFine setter
5710 :
```

```
5720                                    REMark saves new date parameters
5730 DEFine PROCedure date_save
5740   AT 14,0 : CLS 2 : AT 18,1 : PRINT "saving new date ..."
5750   DELETE boot_dev$&"boot_last_date"          : REMark delete old file
5760   OPEN_NEW #3,boot_dev$&"boot_last_date"     : REMark open new file
5770   PRINT #3,"9000 DEFINE PROCedure last_date" : REMark write procedure
5780   PRINT #3,"9010   let year=";year;":let month=";month;":let day=";day
5790   PRINT #3,"9020   let hour=";hour;":let minute=";minute;":let second=";0
5800   PRINT #3,"9030   sdate year,month,day,hour,minute,second"
5810   PRINT #3,"9040  END DEFine last_date"
5820   CLOSE #3
5830  END DEFine save_date
5840 :                                  REMark display time/date cleary in am/pm
5850 DEFine PROCedure display_date
5860     CSIZE 2,1 : LET d$ = DATE$ : LET t$ = d$(13 TO) : LET d$ = d$(1 TO 12)
5870     LET t = t$(1 TO 2): LET p$ = " am": IF t > 11: LET p$ = " pm" : END IF
5880     IF t > 11
5890       LET p$ = " pm" : IF t > 12 : LET t$ = (t-12)&t$(3 TO) : END IF
5900       IF LEN(t$)<8 : LET t$ = " "&t$
5910      END IF
5920     IF t$(1)="0" :  LET t$ = " "&t$(2 TO) : END IF
5930     AT 4,19 : PRINT DAY$ : AT 4,4 : PRINT d$; : AT 4,27 : PRINT t$;p$
5940     CSIZE 0,0
5950  END DEFine display_date
5960 :                                  REMark makes centered screen headings
5970 DEFine PROCedure heading(head$)
5980   CLS #0 : CLS #1 : CSIZE 2,1 : PAPER 7 : INK 0 : AT 0,0 : CLS 4
5990   AT 0,20-LEN(head$)/2 : PRINT head$ : CSIZE 0,0 : PAPER 0 : INK 7
6000  END DEFine heading
6010 :                                  REMark prompts until y(Y) or n(N) typed
6020 DEFine FuNction yorn(l,q$)
6030   REPeat loop
6040     AT l,0 : CLS 4 : PRINT q$;" [y/n] : "; : PAPER 7 : INK 0 : PRINT " ";
6050     LET a$ = INKEY$(-1) : PRINT a$!;  : PAPER 0 : INK 7 : CLS 3
6060     IF a$=="y" : RETurn 1 : END IF
6070     IF a$=="n" : RETurn 0 : END IF
6080    END REPeat loop
6090  END DEFine yorn$
```

BOOT_LAST_DATE **program 3**

```
9000 DEFine PROCedure last_date
9010   LET year=1989:LET month=9:LET day=24
9020   LET hour=6:LET minute=46:LET second=0
9030   SDATE year,month,day,hour,minute,second
9040  END DEFine last_date
```

Comments on programs :

BOOT is pretty straight forward. Lines 1030 and 1040 should make sense for your device and new boot program name. Lines 1170 to 1390 need to also apply to your system tool kit and desires. Remove and alter as desired. You can always skip that section by answering 'n' to the initialize question. At line 1420 the system clock function DATE returns the date in seconds which is compared to the lower and upper bounds set at lines 1100,1110. If the date is not sensible the BOOT_SET_DATE routine is merged. Otherwise you are asked whether the date is correct. After the date is okay you just hit <ENTER> to start the SUPERBASIC program set at line 1040 or you can enter something different.

BOOT_SET_DATE is set up so it will run as an independent program. You may have to run it that way if the BOOT_LAST_DATE file gets lost. Normally it will be merged with boot when needed. Of interest is the call from **date_set** to **setter** for each part of the date. The way SUPERBASIC works is that the variable 'minute' gets passed to 'value'(see how they correspond in the argument list). The value of 'minute' in **date_set** upon return is whatever 'value' was set to in **setter**. I use some slicing and concatenation to get the date in the desirable form in **display_date**. Procedure **date_save** is interesting in that it create a file writes a procedure to it that can merged to boot if needed. This is a technique that can be very usefull.

BOOT_LAST_DATE is written by BOOT_DATE_SET except the first time when your write it.

QL NETWORKing

by Al Feng

Claims abound for computer hardware and software. The original claims for SINCLAIR's QL (aka ZX-83) promised the capability of a PEER/PEER NETWORK because the QL had been designed for use in a "business" environment. The two 'NET' ports bear witness to this intent. In a PEER/PEER NETWORK each QL would have the capability of sharing data and devices equally (therefore, as a "peers").

Reality interferes with claims (design?) and a standard QL is a stand alone computer. It is generally conceded that there are limitations in the standard QL firmware (i.e., QDOS) which makes PEER/PEER NETWORKing difficult, if not impractical and/or impossible. This may yet be a matter of poor communication by those who have claimed to have NETWORKed several QLs together using 'NETi(n)' and 'NETo(ut)' in the way of bad or incomplete examples. Regardless, I have tried many attempts at establishing a NETWORK using two QLs using 'NETi' and 'NETo' to no avail.

Regardless, it is possible to NETWORK several Sinclair QLs via a dedicated FILE-SERVER/CLIENT NETWORK. This is made possible by the multi-TASKing capabilities of the QDOS operating system with the assistance of SuperTOOLKIT coding on each participating QL (both authored by Tony Tebby). Occassional explanations of how this has been performed has been published in various magazines and newsletters. Most have been oblique.

At a recent meeting of our local user group (Chicago Area Timex User Group / CATUG) a QL NETWORKing demonstration was performed for the benefit of those who hadn't seen it before. Constraints of physical space precluded linking more than two QLs together; but, as many as 63 CLIENTs may be "serviced" by a single SERVER.

There are practical constraints to employing any NETWORK. If the FILE-SERVER is pre-occupied with another CLIENT then access will be delayed until the first CLIENT is serviced. The more CLIENTS attempting access, the slower the service. The specifications indicate that as many as eight FILE-SERVERs may be active on a QL NETWORK at any given time.

In a QL NETWORK, each computer active on the NETWORK is assigned a 'NET(work) number' (e.g., 'net 1'). NETWORK 'numbers' are assigned arbitrarily, with the only proviso being that two QLs not have a duplicate 'NET numbers.'

It is generally espoused, based on "other" LOCAL AREA NETWORKs (LAN) implementations, that the QL with the most hardware (e.g., a hard disk, floppy disk, printer, etc.) attached act as the FILE-SERVER. The "first" computer (Butch's) was fully configured with a TRUMP card running dual 720K 5.25" disk drives. The "second" computer (Steve's) was a 128K QL to which a SuperTOOLKIT eprom was attached via the 'ROM' port.

To implement the NETWORK the following commands were typed into each computer ('<ENTER>' indicates that the 'ENTER' key was pressed after entering the preceding statement):

```
(BUTCH's QL)          (STEVE's QL)

NET 1  <ENTER>        NET 2        <ENTER>
FSERVE <ENTER>        NFS_USE flp, n1_flp1_, n1_flp2_ <ENTER>
```

The "FSERVE" statement which is entered indicates that this QL's function within the NETWORK is to act as a FILE-SERVER.

The multiple statement INPUT into the CLIENT QL indicates that you want any calls made to using 'flp1_' and 'flp2_' be answered by the computer which was assigned 'NET(work number) 1' by the devices specified in the second and third part of the statement. In this instance, we wanted a call to 'flp1_' to be answered by 'flp1_' on the QL wich was assigned a 'NET(work number) 1'; and similarly, any calls to 'flp2_' by the client would access 'flp2_' of the QL whose address was 'NET 1'. If a disk drive interface had been attached to Steve's QL, then it would have been made dormant by the "NFS_USE flp" statement.

The computers were connected with a shielded cable having 1/8" microphone jacks (included with the QL) via the NET ports located next to the POWER socket. If we had wanted to connect more QLs, we would have indicated a 'NET number' (eg., NET 3) for the CLIENT followed by the NFS_USE statement. Alternatively, we could have accessed Butch's RAM disks, or microdrives (limited to a choice of two of any combination).

After establishing his QL as the FILE-SERVER, Butch loaded in a copy of the Psion XCHANGE. A copy of QLSS from one of Butch's disks was LOADed into Steve's QL.

LOADing via the NETWORK was much slower than direct (dedicated) access to the disk drives. There are at least two reasons for this. First, communication is done at a rate of "only" 19,200 baud. And, secondly, the transmission is not continuous because the SERVER is periodically checking for "other" input. That is, the NETWORK is being multi-TASKed along with SuperBASIC and whatever else the user may have EXEC'd on the FILE-SERVER.

Multi-TASKing tends to slow effective program/TASK speed. In part, this is because each TASK is given a priority number between 0 (off) and 127 (maximum), with 32 being the "standard" priority given to either SuperBASIC or an EXEC_W'd program. It appears that background TASKs are suspended by the operating system.

While the "server priority" appears to be '8', the SERVER must constanty monitor the NETWORK for any communication from CLIENTs and allow access to its I/O storage devices. In effect, we "disabled" Butch's QL while we were accessing his drives; and, it seemed to be moderatlely "crippled" for normal use.

NETWORKing on the QL clearly has some drawbacks. But it should be noted that this "crippling" effect occurs in all LOCAL AREA NETWORKs. In the non-SINCLAIR world, a single computer is often dedicated to the TASK of being the FILE-SERVER (eg., a mainframe, a minicomputer, or 80286/i386DX/i486). This should be taken as the model for QL NETWORKing, also, whereby a single QL is dedicated to the task of acting as the FILE-SERVER, with other QLs acting as CLIENTs. This

becomes a valid application when several computers wish to share a common source of data (i.e., from a hard-disk). Use of a NETWORK as illustrated appears to be impractical for "home" applications.

The following SuperBASIC program may facilitate implementing multiple CLIENT stations in a FILE-SERVER/CLIENT NETWORK:

```
10 WINDOW#0,448,52,40,204: PAPER#0,7:INK#0,0
20 WINDOW#1,512,256,0,0: WINDOW#2,448,226,40,30
30 CLS: PAPER 7: INK 0: INK#1,3: CSIZE 0,0
40 PRINT#0\,'[ ';VER$;' ROM ] '; \, FREE_MEM;' bytes'
50 CSIZE 2,1
60 INPUT,\\\\\\,"NETWORK STATION NUMBER:  ";a$: NET a$
70 NFS_USE flp, n1_flp1_, n1_flp2_
80 CLS: INK 37: PRINT,\\\\\\, "NETWORK";: INK 0:
       PRINT " CLIENT STATION";:  INK 2:PRINT" ( ";:INK 0:
       PRINT a$;:INK 2:PRINT " )"
90 FOR n=0 TO (a$-1):BEEP 250,10 :PAUSE 8:NEXT n
100 REMark ******* CLIENT (PLATYPUS Software 1988) *********
```

SAVE this program as either BOOT or CLIENT_BAS. If you have a hard-disk, then you may wish to change line 70 to read:

```
70 NFS_USE flp, n1_flp1_, n1_hdk1_          (?)
```

to access the hard-disk when a call to 'flp2_' is made by the CLIENT. Line 90 provides an audible confirmation of the station number. The program "terminates" with the active cursor in 'window#0'. Either LOAD your regular boot when this appears, or add a line which reads:

```
95  LRUN/EXEC_W flp1_ProgramName
```

where "ProgramName" is the name of the program you wish to LOAD at the given CLIENT station (substitute 'EXEC_W' for 'LRUN' as necessary).

FUDGING TO GET A PEER/PEER NETWORK

Since up to eight QLs can theoretically perform the FILE-SERVER role within the NETWORK, it is possible to have PEER/PEER NETWORK of sorts by INPUTting both the NFS_USE and FSERVE statements on each QL. What you may end up with is 'dueling devices', and a potential interruption of primary keyboard INPUT while someone attempts to access one of your devices.

Screen enhancer code (I prefer SPEEDSCREEN) is recommended if you plan to use the FILE-SERVER as an active terminal.

"HOME" USE OF QL NETWORKing

Okay. You've got two or more QLs and you've got SuperTOOLKIT coding on at least one of them. But, you don't have a hard disk (who does?). The $64 question is:

> Is NETWORKing practical enough for me to consider getting a second SuperTOOLKIT?

If you don't even have a SuperTOOLKIT, is a possible expenditure of more than $100 for the NETWORKing capability money well spent? Just how much money you should spend on an orphan computer depends on how much viability you feel the operating system has for you individually. Had Sinclair Research not folded-up its tent, the price of the 128K QL would probably be more expensive than the

current $100 (at the time of this writing). I suspect that it would have been plausible that the ZX-84/85, or at least the ZX-86 would have had an extended operating system which would have incorporated some of the SuperTOOLKIT coding with the 128K QL persisting as do the 8088/V-20 IBM PC clones.

Most people who have SuperTOOLKIT coding (originally available as "code" which was LRESPRed into RAM as well as the standard EPROM, it is now provided on several disk drive interfaces) claim that they wouldn't live without out it. What does SuperTOOLKIT coding provide besides the facility to expedite a FILE-SERVER/CLIENT NETWORK? It does make file manipulation and SuperBASIC editing more convenient; but, it is not a necessity.

If you already have the hardware [or, extra cash to spend!], then you can indeed set up a viable NETWORK for "home" use if one ignores the conventional wisdom for setting up a LAN. What you want to do is to utilize the FILE-SERVER only as a source of additional memory (as with "extended memory" for MS-DOS) which is utilized to provide extra RAM storage via RAM DISKs.

If the INPUT into the two QL's inicated in the first part of this article had been reveresed then the QL with the TRUMP card would have been the CLIENT, and the role of the FILE-SERVER would have been taken by the 128K QL.

```
[ 128K QL ]          [ TRUMPed-up QL ]

NET 1  <ENTER>       NET 2        <ENTER>
FSERVE <ENTER>       NFS_USE mdv, n1_ram1_, n1_ram2_ <ENTER>
```

In this example, since the 128K QL did not have disk drives, calls were made to theoretical RAM DISKs. Calls to 'mdv1_' and 'mdv2_' are now answered by 'ram1_' and 'ram2_' of the QL with the 'NET 1' assignment.

Because a standard QL does not have RAM DISKing capabilities, extra code needs to be LRESPRed. While this can be purchased, code to provide "static" RAM DISKing is also available with the CHOice program code; and, "dynamic" RAM DISKing is available with the QRAM program code (a STATIC RAM DISK needs to be formatted whereas a DYNAMIC RAM DISK does not). A 128K QL can provide a STATIC RAM DISK of about 80K, or 160 sectors.

This doesn't sound like a lot, and it isn't. But, if you have any additional memory expansion beyond what you are currently using on your primary QL than that which QDOS is capable of directly addressing then it can be implemented by the FILE-SERVER for the STATIC/DYNAMIC RAM DISK.

A multiple QL configuration utilizing twin 512K Expanderams is currently being run as I input this text to ensure viability of the concept described. Essentially, a second QL with a 512K Expanderam was powered up in addition to my normal QL which has an 512K Expanderam and Cumana disk interface. The second QL was designated as the FILE-SERVER. My regular QL was designated as the CLIENT. The initial statements were as follows:

```
[ 512K QL ]          [ 512K QL with CUMANA I/F, etc. ]

NET 1  <ENTER>       NET 2        <ENTER>
FSERVE <ENTER>       NFS_USE mdv, n1_ram1_, n1_ram2_ <ENTER>
```

DYNAMIC RAM DISK code used on the FILE-SERVER in this instance.
If I had a third QL with a SuperTOOLKIT, it too could have been designated as a FILE-SERVER with its memory being utilized as a RAM DISK:

```
                     NET 2        <ENTER>
                     NFS_USE mdv, n1_ram1_, n3_ram1_ <ENTER>
```

If the third QL also had a 512K memory expansion, then it this hardware configuration would provide two RAM disks of about 547K each (1094k total) in addition to whatever memory was available to the primary QL! However, before you run to the phone to order additional hardware one should note that this may only be viable if one already has the extra QL, extra memory, et cetera. I suspect that configuration is NOT price competitive with the MEGA-board. Again, remember that you are limited to using the "extra" memory as RAM disks.

There is an advantage to utilizing "external" RAM disks. If the program you are working locks up the keyboard (it happens), you will not loose the contents of the data which had been stored in RAM when you RESET the computer. Of course, programs and data stored in the RAM of the primary computer will be lost; and, power spikes or outages will result in a loss of the contents in any RAM disk.

Two things are clear from this type of memory expansion.

The first is that one has to get used to making 'calls' to 'mdv1_' or 'mdv2_' for the extended memory. This isn't too difficult; but, if you opt for this type of system configuration as a permanent set up, then you may want to modify a majority of your programs to look to 'mdv1_'/'mdv2_' for data storage. [The DYNAMIC RAM DISKs on the primary QL are still referred to as 'ram1_'/'ram2_']

The second mistake I notice myself making is that if I "walk away" for a moment and come back, I often begin typing on the "wrong" keyboard. This is more a matter of logistics, and can be resolved by a number of options, including placing the FILE-SERVER behind the CLIENT, or simply 'out of reach'.

It appears that a CLIENT cannot assign one of its own storage devices to the NFS_USE statement as in the following example:

```
[ any QL ]              [ any other QL ]

NET 1  <ENTER>          NET 2          <ENTER>
FSERVE <ENTER>          NFS_USE mdv, n1_ram1_, n2_mdv2_ <ENTER>
```

It also appears that "NFS_USE ram" is does not appear to be a valid configuration. A request for another QL's RAM would seem to require either an "NFS_USE mdv" or an "NFS_USE flp" statement, with similar devices then become inoperative on the CLIENT.

The NFS_USE statement appears to have some limitations which are not flagged by the system. In fact, I have found that there is more flexibility than had been immediately evident. First of all, the NFS_USE statement made be made to request just about ANY designation. You can theoretically use the statement:

```
NFS_USE bozo, n1_ram1_, n1_ram2_
```

The difficulty with a four letter device name is that you may CRASH some programs which are only looking for a three character string. So, the above "name" should be truncated to "boz" or "bzo" or any other three character device name. Of course, you can use a more reasonable name such as "fdk", "rdk" (?). "Net" might be a good choice, but existing keywords are invalid. Remember the assigned designation so you may readily access it.

The NFS_USE statement may be changed (several times) to access other devices from other FILE-SERVERs as needed.

I don't have a current application for utilizing this much memory since I "learned" not to multi-TASK more than two programs at any given time; and, it may take me a while to get in the habit LOADing more than I need. Of course, some people who multi-TASK their programs prefer to LOAD all of them in at once; and, some programs are memory hungry (e.g., EXCHANGE).

HAPPY TRAILS, AND COMPUTING, TO YOU. . .

STRING SLICING, COERCION & CONCATENATION

Peter Hale, P.O. Box 8763, Boston, MA 02114

String concatenation is a subject that can't easily be covered in a vacuum. It must include string slicing and coercion as well.

A good starting point on those topics is in the Concepts section of the QL manual. See coercion on page 11, data types and variables on page 13, slicing on page 46 and string arrays/variables on page 51.

What follows here integrates those subjects.

A SuperBASIC (or Archive) string can have up to 32766 characters. It can be stored as a variable if the variable has the suffix $.

Any of the 256 characters in the QL character set can be part of a string, although non-printing characters must be entered as code. Whether entered as code or as a printing character, each character requires only one space in the string.

In SuperBASIC each character in a string is located by its distance from the first character. In the string a$="abcdedghij" the letter "a" is in first position, the letter "j" is in tenth position. As in other Sinclair BASICS, the letter "a" is found by a$(1) and "j" by a$(10).

Other, less friendly BASICS, make you use MID$, LEFT$ or RIGHT$ functions depending on the character's position in the string.

SuperBASIC string slicing is elegantly simple. The syntax of string slicing is unchanged whether the character is in first, last or mid position.

Getting more than one consecutive character out of a string is similarly straightforward and simple. The syntax is a$(m TO n) where m is the position of the first character and n is that of the last. Thus:

a$(1 to 3) gives "abc"
a$(3 to 6) gives "def"
a$(7 to 7) gives "g"
a$(8 to) gives "hij"

When selecting the balance of a string it is unnecessary to specify the position of the last character.

COERCION

A handy feature of SuperBASIC is coercion. This has nothing to do with arm twisting; it means that a number can be represented as a numerical variable or a string variable depending on the kind of variable defined, and not on whether or not there are quotes.

If you set the variable b=12, it will be stored as a number variable. On the other hand, if you define the variable b$=12, it will be stored as a string variable just as if you had written b$="12".

You can move back and forth through the magic of coercion; that is, b=b$ changes the string variable b$ into a number variable and vice versa.

The restriction, of course, is that you must deal with numbers. Try letting a=a$ in the above example and you crash. It is also a limitation that numbers are stored accurately only to eight (8) significant digits because it is all done in floating point.

(Floating point - a digression: This means that the number is stored in two bytes as a number raised to the power of 10. The second byte is the power of ten to which the first byte is raised.

Thus, b=123456:b$=b returns b$ as "123456". Longer numbers are handled differently. c=1234567:c$=c returns c$ as "1.234567E6"; d=12345678:d$=d returns d$ as "1.2345678E7"; e=123456789:e$=e holds e$ as "1.2345678E8".

Clearly, b$ and b look the same. c$ and c and d$ and d are both accurate, although they look different because the last two characters are the power of ten to which the preceding numbers are raised. e$ and e differ because the ninth digit is truncated.

The moral for using coercion is to stay with numbers under 9 digits; for clarity, stay under 7 digits.)

Coercion tolerates a limited mix of numbers and text, but does not like text.

If f$="1234A", coercion allows the instruction f=f$, but returns f=1234. It will not permit coercion of g="A1234" since the first

character is not a number.

CONCATENATION

The struggle through both string slicing and coercion helps to understand the full potential of concatenation in the Sinclair QL.

SuperBASIC joins two strings to make a third with the operator & (Concepts page 35).

Whenever SuperBASIC sees the operator &, it knows to treat whatever variables are involved as strings.

Thus x$=a$&b$ or x$=a$&b will each return "abcdefghij123456".

If the operator is +, SuperBASIC knows that it should to do an arithmetical operation.

Whether you write x=b+c or x=b$+c$ or x$=b$+c$, the answer for either x or x$ will be 123456 + 1234567 or 1358023.

x=a$+b or x$=a$+b will crash since the first character in a$ is non-numeric ('a').

Really, that's all there is to it, although some examples (using the variables in the text above) may help.

1. a$(8 to 9)&a$(4 to 5)="hide"
2. a$(2)&a$(9)&a$(7)="big"
3. b$+b$=246912
4. b$&b$="123456123456"
5. a$(2)&b="b23456"
6. b$(4)&b$(2)="42" but b$(2)&b$(4)="24"
7. b$(4)+b$(2)=6 or b$(2)+b$(4)=6
8. b$(2 to 3)+b$(4 to 5)=68
9. b$(2 to 3)&b$(4 to 5)="2345"
10. y$=a$(1 to len(a$)-4): z$=b$(4 to 5)
    y$&z$="abcdef45"
11. if x$=a$&b$="abcdefghij123456" then
    x$(4)&x$(9)&x$(7)&" "&x$(12)&" "&
    x$(8 to 9)&x$(4 to 5)= ?

Just don't try b(4)+b(5); number slicing doesn't work. Coerce first, then be gentle, applies to SuperBASIC, but not to life in general.

THE QL'S FORBIDDEN BYTE

-- Bob Hartung --

Those of us who "cut our computer teeth" on the ZX80/81 learned quite early about the caveat that accompanied any use of CHR$ 118 when machine code was stored in REMs. Like the Cheshire cat, the line could disappear right before your eyes till nothing was left but the grin! The problem was that this particular character was used to mark end-of-line in memory as well as being the new-line (ENTER) character.

Shades of de-ja vous, the QL has done it again with CHR$(10), but the matter is further complicated by the manner in which data saves/loads are done. A case in point was brought to my attention by Tom Sutton when he keyed in the Mail File listing (Oct. 88 UPDATE). Everything worked fine until he saved and loaded back a data file in which CHR$(10) happened to exist in the 115th byte of a string array element used to store the length of first-name entries. The data became corrupted at that point during the save/load processes.

Without doing an extensive re-write of Mail File, DATA_FILE, DATA_FILEQ, and QSO_FILE which all use this kind of approach to storing the lengths of first-entries, a quick and dirty fix is to simply change this CHR$(10) to CHR$(11) whenever it occurs. This will make no difference when fields one and two are printed out on separate lines, and will add only one extra space between a last name and a first name 10 characters in length when printed out on the same line.

The following lines may be inserted in these listings to make this correction:

```
MAIL FILE:
825 IF CODE(f$(n,115))=10 THEN LET f$(n,115)=CHR$
(11)

ADDRFILE1:
695 IF CODE(f$(n,179))=10 THEN LET f$(n,179)=CHR$
(11)

DATA_FILE and DATA_FILEQ:
805  IF  CODE(f$(n,cs))=10 THEN LET f$(n,cs)=CHR$
(11)

QSO_FILE and QSO_FILE2:
1005  IF CODE(f$(n,cs))=10 THEN LET f$(n,cs)=CHR$
(11)
```

If you have versions of these or any other of my issue disk/mdv programs that you would like to have updated, please send return postage along with a copy of the program(s) on mdv or 5¼" DSDD disk compatible with TRUMP CARD.

(DATA_FILEQ has pre-defined fields and labels for use as a logbook according to formal protocol followed by many ham radio logbooks)

PSION XCHANGE

Bill Woodward

Last month I purchased the QL XCHANGE from Sharp's. For a long time I had wondered when and if PSION XCHANGE would be made available to us American orphan QL owners. It is one of the best $59.95 purchases I have made. The four PSION programs are tied together in the XCHANGE with the ability to share information between the programs without going to all the hassel of changing mdvs or disks or importing and exporting. The more I use XCHANGE the more user friendly features I see and the better I like it.

There are several tutorials on the disk that teach you ABACUS & EASEL. They take you one step at a time show you how to proceed. These tutorials are loaded from Xchange using the command key F3; t (for tsl or Task Specific Language); the file name, e.g., abacus1_tsl, abacus2_tsl, abacus3_tsl, easel1_tsl, or easel2_tsl; ENTER and watch the tutorial unfold. ABACUS1_tsl displays the spreadsheet first and proceeds to automatically enter an animation of CASH FLOW MODELLING from page 18-19 of QLUG. ABACUS2_tsl & ABACUS3_TSL displays the spreadsheet and tutors a simple financial model which EASEL1_tsl & EASEL2_tsl model on the screen.

Xchange F3 has 11 commands, i.e., BACKUP,CONTINUE,DELETE,LIST (dir), NEW,OUTPUT(background printing), QUIT, RENAME, SET (default drives like CONFIG but easier to use) TSL, & XCHANGE(swaps data between files) which are accessed with F6 and you can return to your present task or a new one.

From XCHANGE you access superbasic with CTRL C and run INSTALL_ BAS and set your printer parameters. I spent several days setting up my printers. I have both serial and parallel. The printer_dat is held in xchange_dat. So I set up my serial printer for continuous paper in xchange_dat. COPY flp1_xchange_dat,flp1_ cprinter_dat. I then LRUN flp1_install_bas and changed continuous to no and set up single sheet paper in xchange_dat. I am doing the same for my parallel printer. The data will be psprint_dat and pcprint_dat.

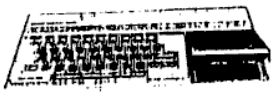
When writing letters I use single sheets so in XCHANGE I set default to FLP2_ while I am writing. When I need to print, I hit shift F1 to get into XCHANGE; F3, S, flp1_ to default for printer. If I wish to use any other feed, I use F3,S,ram1_ for default and CONTROL, C to get into superbasic, and type COPY flp1_cprinter_ dat,ram1_xchange_dat if I am using serial printer with fanfold paper or COPY flp1_psprint_dat,ram1_xchange_dat or copy flp1_ pcprint_dat,ram1_ xchange_dat for my parallel printer. All my printer_dat files have the 10 translate positions filled, too, so I can use all the features of my printers.

Xchange is like getting a PSION TOOLKIT II. The HELP files are greatly expanded also. for someone like me that is blundering my way along it is a blessing to know help is only F1 away.

PRINTER INTERFACE

Bill Woodward

Recently I acquired a Miracle Systems CENTRONICS PRINTER INTERFACE. it allows you to send the serial output of the QL to either SER1 or SER2 and run a parallel printer. You can run a Dot Matrix from SER2 and a daisy wheel printer from SER1. One can be serial and the other parallel or by using two Centronics interfaces both can be parallel.



ANOTHER APPROACH TO EXTRA MEMORY BANKS

Larry Crawford, 357 Reyolds Rd, London Ontario Canada N6K 2P8 (519) 657-9119

Back in '86 I was faced with a problem with memory. I had a program in which I stored and sorted movie titles, updating it each month. As the list approached 1000, it was obvious that RAM would soon be full and extra banks of memory were needed. I put together a board using eight 6264 SRAMs to give me 64K in two banks. The following year the 43256 chips became available so. I redesigned the circuit to give me 256K in eight banks. This article will describe how it works.

One of the problems associated with adding memory is the necessity of staying out of the way of the system operating routines. Memory is configured into eight chunks of 8K each (page 2, Tech Manual). The simplest way of of avoiding conflict is to leave chunks 0 to 3 in Home Bank enabled at all times and to enable only chunks 4 to 7 of any other banks added. With an appropriate interface, Expansion Banks can be added and controlled through the I/O ports. That is the approach I took. Since over 200 ports are not used by the system or other peripherals, we can add over 6.4 MEGAbytes of RAM! Let's start less abitiously and deal with ports 0 to 8 only.

HOW IT WORKS

When the OUT command is used in BASIC, or the OUT (C),A instruction in M/C, the number of the I/O port is put onto address lines A0 to A7 then the IORQ and MR lines go LOW simultaneously (page 12, Tech Manual). When that happens, two of the three ENABLE inputs of the 74138 decoder are set correctly. The third one has yet to be considered. Since the ports we want to use are 0 to 8, address lines A4 to A7 will be low. Consequently A4, inverted by one of the NAND gates in U1, applies the third ENABLE signal to the 74138. Up to this point, all outputs of the decoder have been HIGH. Since lines A5 to A7 are LOW and applied to the three address inputs, the output 0 pin will go LOW momentarily.

This applies a pulse to the Clock Pulse pin of the 74374 octal flip-flop which then latches the outputs from the 7442 decoder. While all of this was happening, address lines A0 to A3 (carrying the port number in binary form), were decoded to a one-of-ten output by the 7442. Only the active output goes LOW: the others remain HIGH. These outputs are the ones latched by the flip-flop.

The eight outputs 1 to 8 of the flip-flop are applied to eight inputs of two 7432 OR gates. It is at this stage that the final address decoding is done. Address line A15 is applied to the other inputs of the OR gates through an inverter. Since the extra Banks are located in the upper 32K of memory, A15 will be HIGH when they are active. Thus only when one of the flip-flop outputs is LOW AND A15 is high will one of the OR gate outputs be LOW. These outputs are connected to the respective ENABLE (LOW active) pins of the DRAMs.

A very useful "frill" is the inclusion of 8 LEDs to indicate which bank is active. Should any bank except the Home bank be active, all sorts of strange things can happen unexpectedly. For instance, it is not possible to LOAD or SAVE when another bank is active. It is also very useful to know exactly which bank is being used by your routines and to know whether or not things are going along as you had planned.

Another option is the provision of a battery backup to provide non-volitile memory. The data in the bank will be held even after the computer is switched off. The four inputs of the U1 NAND gate that are tied together are connected to the computer's 5V supply through an RC circuit which will prevent the output of the gates from going LOW for about a third of a second, thus allowing the power supply a little time to settle down. This resistor and capacitor and the eight pullup resistors connected to pin 20 of the memory chips would not needed if battery backup is not wanted.

BANK SWITCHING

Port F4 (244 decimal) contains the Horizontal Select Register (page 37,

Tech Manual). The 8 bits in this register determine if the respective chunks of the Home bank are active. Thus to "make room" for another bank in chunks 4 to 7, we must turn off these chunks by changing the Register to 11110000 binary (240 decimal or F0 hex). The command OUT 244,240 or the M/C instructions <LD A,F0> <OUT (F4),A> will do the job. Now, to activate bank n, the command OUT n,0 is used. Where n is an integer from 1 to 8. The 0 in this command can, in fact, be any number because the data it represents is not decoded by the interface. Some value must be given to satisfy the System logic. To return to normal Home bank operation, use OUT 0,0 : OUT 244,0. Bank 0 is the Home bank since output 0 of the 7442 decoder is not connected to any of the SRAMs.

CONSTRUCTION

After several frustrating attempts at etching a board using a photo resist, I decided to go the hand-wiring route.

Use a 4"x4" piece of perfboard with 0.1" centered holes and a strip of old circuit board with double sided edge connector fingers glued to one edge. Mine came from a board in a surplus store. If you have to make your own, it is prudent to tin plate the copper fingers to head off oxidation problems later. The Liquid Tin to do this is available from large suppliers such as Active Components.

Mount the sockets by feeding their legs through the perfboard holes and bending them flat against the board on the other side. Use wire-wrap wire to go from point to point, starting with one of the edge connectors. It goes surprisingly quickly since it is possible to melt through the insulation on the wire to make a connection without having to cut and strip the ends. The multitude of wires running between the RAMs on the left and right halves of the board can be threaded through holes and cross over on the top side, thus getting them out of the way. Check all lines with an ohmmeter before plugging in the ICs. After protecting the edge connectors with masking tape, spray the back of the board with clear lacquer to hold the wires in place. The whole project took about 10 hours and plugs into an Oliger expansion board.

Cont Next PAGE

PUBLIC DOMAIN LIBRARIES

And some new Ideas

The CATS TSUG advises that their library for the TS-2068 is on Cassette only. Three tapes are available with over 140 programs total. All three for $9.00 or $3.00 each. Be nice and send an extra buck for postage. Joe Miller CATS, 1704 Dayton St., Hyattsville, MD 20703.

ALL SYSTEMS LIBRARY: The Ontario, Canada TSUG has a broad Public Domain Sinclair Library, too diversified to list in detail. Covered are Cassette for the TS-2068, an extensive LARKEN DISK Library, and now a Sinclair QL library is being established. For details send SASE with two US Dollar bills for information. Save your US Stamp as Canadian Post needs their own revenue for postage. And dont send US coin or stamps. Returned will be a listing of their library and low prices. Its worth your buck. George Chambers, 14 Richome Court, Scarborough, Ontario M1K 1Y1, Canada.

SINCUS LIBRARY: TS-2068 Library on Cassette, Oliger Disk, or LKDOS DISK. Send $2.00 for information package to: John Colona SINCUS, 28 Guilfoyle Ave., Binghampton, NY 13903.

VISTA LIBRARY: TS-1000 and TS-2068 on Cassette, and TS-1000 and TS-2068 on LKDOS DISK. Send $2.00 for information package to: Tony Willing VISTA, Box 199, Vashon, WA 98070.

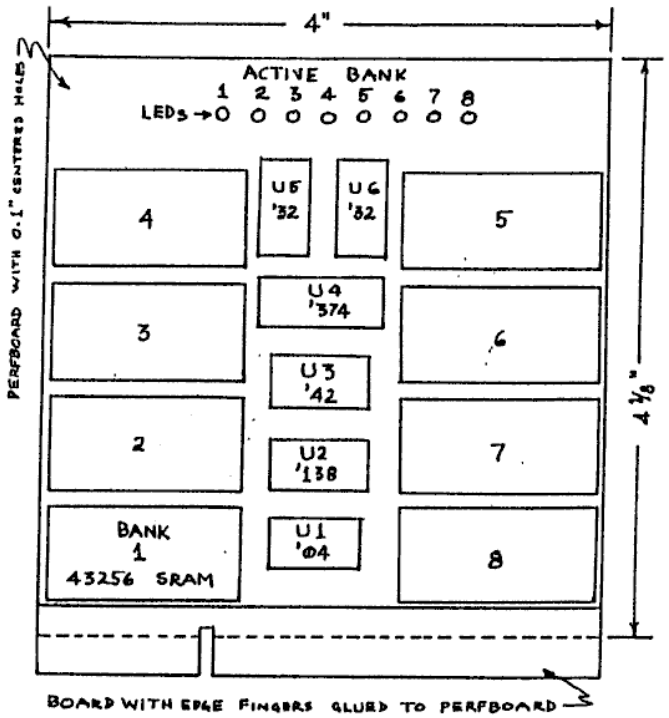
ISTUG LIBRARY: TS-1000 and TS-2068 Library on Cassette and Oliger Disk. Send $2.00 to: Frank Davis ISTUG, 513 East Main, Peru, IN 46970 or call (317 473-8031)

NOTICE TO TSUGS: Those TSUGS not mentioned above that desire public announcement of their services or programs are offerred space in Update free for such announcements.

Editorial Review of Public Domain Libraries

Update has most of the above listed libraries. They came by the bag at ridiculuous low prices, something like 5 cents or less per program. There are enough good and useful programs in every disk or cassette to be worth five times the TSUG price for the library. Each library will have some overlaps but each will have programs that the others do not have. These are put together with a lot of volenteer long hours

Continued Page 43



256 K EXPANSION BANKS COMPONENT LAYOUT
LD CRAWFORD JUL '89

(Cont. Page 46)

(LIBRARIES, From pg 42)

contributed freely, thus the TSUG can be generous, and thus the low prices.

Generally, public domain software are programs and utilities created by individual programmers who donated their work to all. Some are lacking in application while others are quite extensive. About 30% are games. For even the lessor programs there are interesting programming techniques to be studied and used. The libraries on disk generally go beyond this scope and provide exceptionally useful disk management tool kits, and have programs that were designed for cassette opertation and then modified and expanded for disk operation. An example of this is the LKDOS Library by the Ontario TSUG which has an invaluable group of LKDOS Disk Management programs.

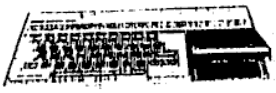
The Sinclair QL is lacking in Public Domain programs available in North America. One group, the Ontario TSUG is working hard to produce a QL PD Library. HELP is needed from programmers who are willing to donate useful utilities to Public Domain. Send your programs on disk or mdv to: Hugh Howie, Sinc Link, 586 Oneida Drive, Burlington, Ontario L7T 3V3, CANADA. Even the most insignificant of utilities will be appreciated. If it does a job for you it will be useful to others. CREDITS will be given to the author.

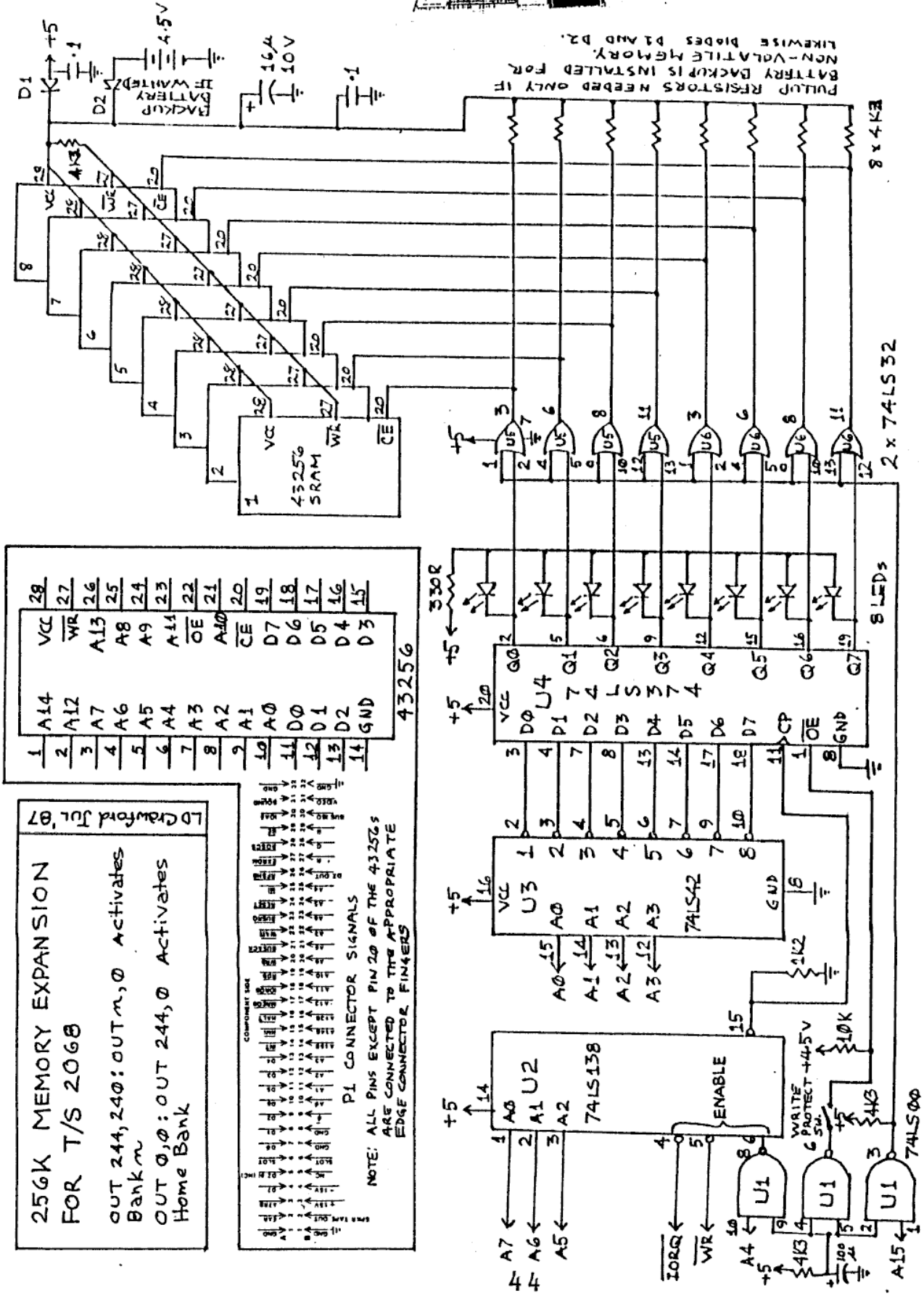
Suggestions to TSUGS

While on the subject of TSUGS and PUBLIC DOMAIN Libraries, I have the feeling that there is a need for an intermediate library effort of sorts in the TSUG programs. How about a "SHAREWARE LIBRARY" for both TS-2068 and Sinclair QL? Many programmers have become dormant because of the low level of buying activity in the Sinclar user group. This is of course because of our low numbers. A TSUG Librarian could rummage through past magazine issues and find dozens of programmers and small software businesses that are no longer active.

I'll bet that letters to these non-active programmers can result in many of their excellent programs being assembled into a "SHAREWARE LIBRARY" that would be even more useful than our public domain libraries. Of course, the librarian would have to craft a "SHAREWARE NOTICE" for every single program so that the Author would have some chance of renumeration from honest users. My visualization of how this would work is: A TSUG

Continued, Page 49




256K MEMORY EXPANSION
FOR T/S 2068
OUT 244,240: OUT n,0 Activates Bank n
OUT 0,0: OUT 244,0 Activates Home Bank
L.D Crawford Jul '87
P1 CONNECTOR SIGNALS
NOTE: ALL PINS EXCEPT PIN 20 OF THE 43256s ARE CONNECTED TO THE APPROPRIATE EDGE CONNECTOR FINGERS
43256
43256 SRAM
74LS42
74LS138
74LS00
74LS32
2 x 74LS32
8 LEDs
330R
8 x 4K3
4K3
2K2
10K
1K
100
.1
16µ 10V
4.5V
BACKUP BATTERY IF WANTED
D1
D2
PULL UP RESISTORS NEEDED ONLY IF BATTERY BACKUP IS INSTALLED FOR NON-VOLATILE MEMORY.
LIKEWISE DIODES D1 AND D2.
WRITE PROTECT SW.
ENABLE
A7 A6 A5
A4
A15
IORQ
WR
44

WINDOW DEFINING:
A Program, part II

by David McCall

In the last article, we wrote the assembly language routine which will become part of our window defining program. In this article we will write and discuss the different functions that will be used. SuperBASIC gives us the ability to write very versatile, many variable functions which can do more than merely return a number. Next time the procedures will be discussed. Once completed, the program will provide all the information discussed below and more in a user friendly pop up window. The window being defined will be able to be positioned and sized on the screen without overwriting anything, such as other windows or text. Every attribute about a window will be defined and a sample window displayed on the screen.

The following SuperBASIC programs, functions and procedures have been written in a very structured style. This is very helpful when trying to follow the logic. Different techniques are used to demonstrate the power and adaptability of SuperBASIC. From looking at the listing, the use of indentation is quite prominent. This is another helpful technique when trying to read long programs because the nested control is very clearly marked by the appropriate indentation.

The description begins with the simple sign function, **sgn%**. It will return the sign of a number: 1 if positive; -1 if negative; and 0 if zero. As indicated by the percentage symbol affixed to the end of the name, the function is of integer type. The logic will be a one lined IF-THEN-ELSE statement, lines 2000 to 2020. First the number needs to be tested against zero and a zero returned if so. Otherwise, the sign which is the absolute value of the number divided by itself is returned.

Notice that more than one RETurn was used. It is important to remember that once a RETurn has been encountered execution leaves the function or procedure at that point. Also, the name of the function is included in the END DEFine statement making it and other functions easier to identify.

The minimum of two numbers is another function which is useful. The logic is given in lines 2280 to 2340. The function name is **Min** and takes two parameters, **mn1** and **mn2**. As indicated by the function and parameters names, it is and takes real or floating point numbers. The long form of the IF-THEN-ELSE statement is used this time. Note that this function can be used to determine the minimum of more than two numbers. For example, the minimum, **x**, of three numbers, **mn1**, **mn2**, **mn3**, can be determined by the following statement:

```
x = Min( mn1, Min( mn2, mn3))
```

All functions and procedures can be called recursively, like above, or within the function or procedure itself.

The next group of functions are all real valued but take no parameters. Instead global variables are used to pass parameters, for several reasons. First, the variables are constantly being changed and second, the functions never require different parameters. Additionally, it shows another alternative for the use of functions. They, **max_xp**, **max_yp**, **max_xpix**, **max_ypix**, **max_xc**, and **max_yx**, determine the maximum x and y point, pixel, and character positions for a window, respectively. Keep in mind that a pixel is the smallest screen element the computer can access, while a point is a position mapped onto these pixels. To clarify this concept, remember that the absolute maximum number of pixels across the screen is 512 while the number of points can be any value depending upon the scaling.

The global variables used are:

scl = the SCALE of the window
xg,yg = the graphics origin defined by the SCALE command
scr_x = the x dimension of the window
scr_y = the y dimension of the window
brder = the BORDER width
cxi,cyi = the number of pixels a character occupies in the x and y directions

The simplest functions is **max_yp**, lines 2040 to 2060. The number of points in the y direction, **scl**, is defined with the SCALE command. The maximum y position is, therefore, the y origin value, **yg**, added to **scl**.

The most difficult to understand function is **max_xp**, lines 2080 to 2100. It is a function of many variables **xg**, **scl**, **brder**, **scr_x**, and **scr_y**, and the computation is complex. The SCALE command scales the window by making the x and y directions appear to have the same interval between points. There are 512 pixels in the x direction and 256 pixels in the y direction. Plus, the actual screen image on your monitor has a ratio of about 802 to 503 (this was determined

experimentally to give the best answer in the function). The number of pixels that are accessible in the window are not **scr_x** and **scr_y** if a border is defined. They are (**scr_y** - 2 * **brder**) and (**scr_x** - 4 * **brder**) -- the BORDER width is doubled in the y-direction and quadrupled in the x-direction.

To calculate the maximum point position in the x-direction, the maximum length of the window in the y-direction must be mapped into length as measured in the x direction. This is done by multiplying **scl** by (503/802). This result must then be multiplied by the ratio of pixels in the different direction, (**scr_x** - 4***brder**)/(**scr_y** - 2***brder**). Finally, it must be added to the x value of the graphics origin, **xg**.

The expression for this calculation given in line 2090 has alternating division and multiplication to maintain as many significant digits as possible. Whenever a calculation requires many multiplications and divisions, it should always be arranged such that the two operations alternate with division coming first. Otherwise more significant digits can be lost by the rounding error in the computer.

As discussed above, the number of accessible pixels in the defined window is (**scr_y** - 2 * **brder**) and (**scr_x** - 4 * **brder**). To obtain the values for **max_xpix** and **max_ypix**, one must be subtracted from both, see lines 2120 to 2180. This makes the pixel origin (0,0) instead of (1,1).

The last window characteristic functions, **max_xc** and **max_yc**, are relatively simple. These determine the maximum number of characters that can be printed in the window with the specified character size. This calculation is the integer portion of the maximum pixels divided by the number of pixels the character takes up. For the x and y directions, the equations are INT((**scr_x** -4***brder** -1)/ **cxi**) and INT((**scr_y** -2***brder** -1)/ **cyi**), respectively. Please refer to lines 2200 to 2260.

Finally, the remaining functions are used in conjunction to replace the standard INPUT. In their present state, they allow only integer values with a specified number of digits and maximum value. The values inputted with these routines are constantly right justified as they are typed in and if nothing is entered, a default value is returned. Also, if positive numbers are only desired, they will ensure this.

The function **get_num%** calls the main function routine, **input%**, which handles all justification and keyboard reading. **Get_num%** performs the limit and sign testing and returns the proper value according to the restrictions given. The first line, 2380, calls **input%** and sets the return value, **old%**, to a default value if nothing has been entered. The next line, 2390, checks to see if the value is greater than the maximum value, **max%**. If greater, it sets the return value to the maximum value. The following line, 2400, checks the positive value only flag, **positive%**, and if set, makes sure the value is positive. Finally, line 2410 returns the value.

The last function, **input%**, lines 2440 to 3420, will not be discussed line by line as the previous ones were because of its length. However, each major section will be discussed and explained. The first five lines initialize the variables, set ink color, and define the local variables. The rest of the function is a REPeat loop with a large SELect ON statement nested inside. Basically the function constantly checks the keyboard with an INKEY$ function and the SELect ON statement performs the necessary tasks depending upon the key entered. The number being entered is kept in a string **stg$**. The variable **pos%** keeps track of the string position the cursor is presently in, **length%** holds the number of digits typed, and **max%** is the maximum number of digits allowed.

The first few lines of the REPeat loop, 2520 to 2560, print the cursor and retrieve the next key. The SELect statement processes the key typed. If the "ESC" key is typed, the program is stopped, lines 2580 to 2590. If "-" or "_" is typed a negative sign is inserted at the far left position, 2600 to 2610. Similarly, if "=" or "+" is typed, a space is put in the far left position. If a number is typed, it is inserted at the present cursor location and the number of digits, **length%**, is increased, 2640 to 2780. As you can see, a double nested IF-THEN-ELSE statement is used. This is necessary because of the different cursor positions, far left, far right, in the middle, that are possible and the string functions in SuperBASIC require different handling in each case.

Lines 2790 to 2850 move the cursor about the already typed numbers and it is not allowed to move across any untyped space. Right deletion is handled with lines 2860 to 3050. Again there are two nested IF-THEN-ELSE structures taking care of the different possibilities that could arise. The outer IF statement checks to see where the cursor is in the string. If it is at the first position, removing a character is easier. However if there has only been one character typed in the string can be reset to its initial state otherwise the rest of the characters need to be shifted.

When the cursor is not in the first position, the string needs to be cut in two and put back together without the deleted character. The **left%** and **right%** variables are used to find the position just left of the cursor and just right of the cursor. The string is then reconstructed after the left position is checked to make sure it is still within the string.

Left deletion is performed by lines 3060 to 3260 and is analogous to the right delete except it does not remove the character the cursor is on like the regular INPUT. The right delete was chosen to perform this action because the routine keeps numbers right justified.

The number is entered whenever "Return", "Up Arrow", or "Down Arrow" is pressed, lines 3270 to 3300. The last ON clause repeats the loop again if any other key is typed. Immediately following the END SELect, lines 3340 to 3350, are the lines which update the string on the screen. Finally, lines 3370 to 3410 return the value entered or -9999, which indicates nothing has been typed in.

Half of the larger window defining program has been covered with this second article. You will notice that the line numbering started at 2000. Each article has been constructed to ensure no line numbering conflicts when they are all merged together at the end of this series. The complete program will be available on an "Issue Disk" also.

LISTING A

```
2000 DEFine FuNction sgn%(t%)
2010    IF t%=0 THEN RETurn 0:ELSE RETurn ABS(t%)/t%
2020 END DEFine sgn%
2030 :
2040 DEFine FuNction max_yp
2050    RETurn scl + yg
2060 END DEFine max_yp
2070 :
2080 DEFine FuNction max_xp
2090    RETurn scl/802*503/(scr_y-2*brder-1)*(scr_x-4*brder-1)+xg
2100 END DEFine max_xp
2110 :
2120 DEFine FuNction max_xpix
2130    RETurn (scr_x-4*brder-1)
2140 END DEFine max_pix
2150 :
2160 DEFine FuNction max_ypix
2170    RETurn (scr_y-2*brder-1)
2180 END DEFine max_ypix
2190 :
2200 DEFine FuNction max_xc
2210    RETurn INT((scr_y-2*brder-1)/cyi)
2220 END DEFine max_xc
2230 :
2240 DEFine FuNction max_yc
2250    RETurn INT((scr_x-4*brder-1)/cxi)
2260 END DEFine max_yc
2270 :
2280 DEFine FuNction Min(mn1,mn2)
2290    IF mn1<mn2 THEN
2300       RETurn mn1
2310    ELSE
2320       RETurn mn2
2330    END IF
2340 END DEFine Min
2350 :
2360 DEFine FuNction get_num%(win%,x,y,len%,old%,max%,positve%)
2370 LOCal t%
2380    t%=input%(win%,x,y,len%):IF t%<>-9999 THEN old%=t%
2390    IF ABS(old%)>max% THEN old%=sgn%(old%)*max%
2400    IF positve% THEN old%=ABS(old%)
2410    RETurn old%
2420 END DEFine get_num%
2430 :
2440 DEFine FuNction input% (win%,x,y,max%)
2450 LOCal loop%,k$(1),k,s$(1),stg$(max%),pos%,right%,left%,length%
2460   INK#win%,7
2470   s$(1)=' '
```

```
2480  stg$=FILL$('_',max%)
2490  pos%=1
2500  length%=0
2510  REPeat loop%
2520    OVER#win%, -1
2530    AT #win%,x,y+max%-pos%:PRINT #win%,'_'
2540    OVER #win%,0
2550    k$(1)=INKEY$(#3)
2560    k=CODE(k$(1))
2570    SELect ON k
2580      ON k=27
2590        STOP
2600      ON k=45,95
2610         s$(1)='-'
2620      ON k=43,61
2630         s$(1)=' '
2640      ON k=48 TO 57
2650        IF pos%=1 THEN
2660           IF max%>1 THEN
2670              stg$(1 TO max%)=stg$(2 TO max%)&k$(1)
2680           ELSE
2690              stg$(1)=k$(1)
2700           END IF
2710        ELSE
2720           IF pos%<max% THEN
2730              stg$(1 TO max%)=stg$(2 TO max%-pos%+1)&k$(1)&
                                      stg$(max%-pos%+2 TO max%)
2740           ELSE
2750              stg$(1 TO max%)=k$(1)&stg$(2 TO max%)
2760           END IF
2770        END IF
2780        length%=(length%+1):IF length%>max% THEN length%=max%
2790      ON k=200
2800        pos% = pos% -1
2810        IF pos%<=0 THEN pos% = 1
2820      ON k=192
2830        pos% = pos% +1
2840        IF pos%>=length%  THEN pos% = length%
2850        IF pos%=0 THEN pos%=1
2860      ON k=202
2870        IF pos% <> 1 THEN
2880            left% = max%-pos%
2890            right% = max%-pos%+2
2900            IF left% < 1 THEN
2910               stg$(1 TO max%)='_'&stg$(2 TO max%)
2920            ELSE
2930               stg$(1 TO max%)='_'&stg$(1 TO left%)&stg$(right% TO max%)
2940            END IF
2950            length%=length%-1:IF length%<0 THEN length%=0
2960         ELSE
2970            IF max%=1 THEN
2980               stg$(1 TO max%)='_'
2990            ELSE
3000               stg$(1 TO max%)='_'&stg$(1 TO max%-1)
3010            END IF
3020            length%=length%-1:IF length%<0 THEN length%=0
3030         END IF
3040         pos%=pos%-1
3050         IF pos%<=0 THEN pos%=1
3060      ON k=194
3070        IF pos% = 1 THEN
3080          IF max%<=2 THEN
3090             IF max%=2 THEN
3100                stg$(1 TO max%)='_'&stg$(max%)
3110                length%=length%-1:IF length%<1 THEN length%=1
3120             END IF
3130          ELSE
3140             stg$(1 TO max%)='_'&stg$(1 TO max%-2)&stg$(max%)
3150             length%=length%-1:IF length%<1 THEN length%=1
3160          END IF
3170        ELSE
3180          left% = max%-pos%-1
```

```
3190            right%=max%-pos%+1
3200            IF left%<1 THEN
3210               stg$(1 TO max%)='_'&stg$(right% TO max%)
3220            ELSE
3230               stg$(1 TO max%)='_'&stg$(1 TO left%)&stg$(right% TO max%)
3240            END IF
3250            length%=length%-1:IF length%<0 THEN length%=0
3260          END IF
3270        ON k=10,208,216
3280          AT#win%,x,y-1
3290          PRINT #win%,s$(1)&stg$
3300          EXIT loop%
3310        ON k= REMAINDER
3320          NEXT loop%
3330      END SELect
3340      AT#win%,x,y-1
3350      PRINT#win%,s$(1);stg$
3360    END REPeat loop%
3370    IF length%=0 THEN
3380       RETurn -9999
3390    ELSE
3400       RETurn s$(1)&stg$(max%-length%+1 TO max%)
3410    END IF
3420 END DEFine input%
```

⟨LIBRARIES (From Pg 43)⟩

Librarian would spend some of the clubs rich finances to write to the programmers and defunct software businesses, explaining the objectives and promising that a Shareware Notice would be attached to any program donated to the library.

The Shareware Library would be assembled and sold for a small fee to re-pay the TSUG treasury and perhaps make a nominal profit. Here we would have a level of professionalism in the Shareware Library that exceeds that of a public domain library. Most honest users woud respond to the Shareware notice and send the author his just deservings for the use of the program. An organized effort to serve the users with good programs could just entice some of our dormant programmers to begin producing again. A TSUG could set its own standards for "Read Me" doc files to enhance the value of the library. Some work in that area would justify the small profits to be realized by the Club treasury.

There is no reason why our active dealers shouldn't pick up upon this idea and develop their own Shareware Libraries. Generally Shareware Software is fully documented within the programming by "Read Me" files, and is not supported by paper manuals. Thus the distribution of Shareware programs is simply one of supplying the programs on a medium. And, the distributor carries no responsibility for adequacy of either the programs or their documentation. The success of the Author in receiving payment for his work depends upon his prowess in explaining his program in the Read-Me files.

If the (honest) user finds the program to be useful, then his conscience works on him to send the Author what the user feels that the program is worth. Here-to-fore the Shareware concept has not been very successful. This is because it has been a singular effort by single progammers who have had difficulty in letting the public know about their product availibility. A combined disk with several author's programs, distributed by a TSUG or Dealer, could be of benefit to all. Update Magazine will help to publicise such an effort by the TSUGS and our Dealers.

⟨MUSINGS From Page 12⟩

other with 30 Mgbt Hard drive and operating MS DOS. Each has two MDV's making seven storage devices at disposal. Anybody want to tackle an algorithm for that scheme?

TS-2068 to QL LINK SOFTWARE? Why not? This is so close to reality that it should be done. What we need is an adaptation of Modem Software that operates direct from the TS-2068's ZSI/O to the QL's Serial I/O port. Another Idea is to use two modems hooked together, and their software, without going through the telephone medium. There are people doing this, but how they do it is not known publically.

I have pile of good stuff for the "Perks and Quirks" column for next issue. There just wasn't room for more in this one. John Juergens accused me of being "anti Micro Drive", which deserves a smart answer, plus more good and better stuff. Incidentally, I use the PSION suite right out of micro drive almost all of the time. I have an idea for a "QL Hard Floppy" for the next issue which I'll get our QL Editor to authenticate.

The UPDATE Computer Desk

Several have ask about how things are set up at the UPDATE Computer room. So, we will use a couple of pages to discribe the lash up. The sketch on the facing page gives a busy diagram of the three computer and two printer system. Not shown is the miriad of power and interface cabling. The whole system is set up on a "door blank", purchased from a lumber yard, varnished to a nice finish. This desk top sits upon three cabinets that have shelves. These teakwood cabinets have ornate picture carvings and were sent home from Thailand during the Viet Nam war. So the entire computer desk is quite attractive.

When such a system is first installed the biggest problem is to avoid getting caught in the spider web of cabling. This requires some planning. Four 6 outlet receptacle boxes are used. Two of them have both voltage spike and RFI protection. The equipment was first placed into position and then the cabling was run to the rear of the desk and lashed up into trunk lines, the surplus lengths folded back and tied. All of the equipment was purchased from our own Sinclair Suppliers over a period of time, which avoided difficulty with cables and connectors not matching.

THE DISK CABINETS: One for the QL has 2 80 track drives. One for the TS-2068 has three 80 track drives and a 40 track. Another for the TS-2068 has two 40 track drives. It is common practice to change drive unit Data Cables while power is on in order to copy disks of different mixtures, 80 to 80, 40 to 40, or 80 to 40. This is quite safe with Shughurt DOS systems. Both monitors sit upon stands that allow equipment to be positioned underneath. Under the QL Monitor is the Coleco Adam Power supply and an A/C receptacle unit. Beneath the TS-2068 monitor is the extension to the I/O bus, the A/B switches, and the Printer Buffer.

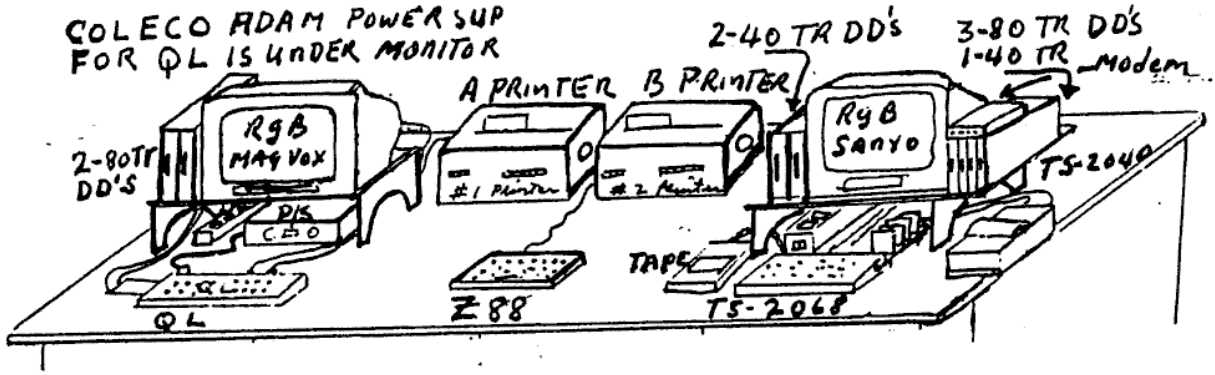
The QL System has Trump Card, EPROM for ROM change, and Speed Screen in ROM. A spare QL has different tool kits than the main. The TS-2068 system is really stacked up, having a minimum system configuration of 344K with the Aerco FD-68 or up to 424K with the combo of LKDOS and Oliger DOS. The main TS-2068 has the (Dan Elliott) modification to provide external power for all external devices connected to the I/O port. This modification was done to the computer itself and a switch added to the rear deck to allow either Timex Power or Coleco Power to the external devices. With power supplied by the Coleco P/S the TS-2068 voltage reg is protected from overload.

The TS-2068 has five Disk Operating systems: 1. Larken LKDOS, 2. Oliger DOS, 3. Aerco FD-68 DOS, 4. MAX DOS, a combo of STACKED LKDOS and Oliger DOS, 5. Stacked Oliger DOS and SPDOS. Actually there is a possibility of another by having the Oliger DOS, LKDOS, and SPDOS in a stacked environment. SPDOS is the "Millenia K" system DOS that resides in RAM.

The PRINTING set up allows either computer to use print buffer and either printer. This allows different print styles to be used at the flip of a switch. Also, the print buffer in this set up allows the output data of three computers to be combined in the print buffer for printing in the same document. Alternate capability exists to hook up two QL's or two TS-2068s as "computers A and B", or to hook up either spare in the position that the Z88 is shown.

TELECOMPUTING: There is a capability for the QL and TS-2068 to transfer ASCII files to each other via two modems and two telephone lines which are terminated at the computer desk. Or, both computers can be downloading files from two different BBS at the same time. Much of the capabilities have langulshed for lack of time to devote to the tasks. While the entire system fits nicely upon the "door top desk", there is need for side board desk top space to store disks and the large library of manuals. So, the computer corner has wrap around desks in a "U" shape. The entire floor space requirement is about 8X10 feet.

<The Desk Top> <TOTAL COST COMPARES With a Medium Equipped IBM XT.>



Did you know that a MAHOGANY FACED door IS cheaper than a 4x8 sheet of plywood? CRAZY, eh?

UPDATE MAGAZINE ISSUE DISKS (MDV)

The Issue Disks contain programs and utilities that are presented in the magazine, plus some longer Softwares that are not given in Update. All of the Sinclair QL programs and some of the TS-2068 programs are authored by contributing programmers. These authors receive 50% of the sales proceeds. The purpose of the Issue Disk program is to encourage our programmers to continue to produce software and utilities for our support. We introduce you to the programmers and hope that you will contact them directly to receive a complete listing of their products.

Update Magazine considers each of the centerpiece titles of these disks to be well worth the price, but we add selected programs to the disks to insure that they are bargains.

Sinclair QL disks are supplied on either Micro Cartridge or 5 1/4" DS DD 80 track Disks. (If for MDV send a Formatted Cart). So far as we know these disks are usable with any QL Disk Interface (CST, Sandy, Trump, Cumma).

TS-2068 Issue Disks are supplied on either 40 or 80 track DSDD 5 1/4", for LKDOS, Oliger DOS, Aerco FD-68.

TS-2068 ISSUE DISKS

1. "DAISY.B6" Disk
A Complete TS-2068 Software System. Data Base Management, Versatile Print Outputs, Disk Management. Over 20 integrated programs.

2. "HARTUNG A" Disk
A complete set of Disk Mgt Utilities for Oliger DOS Only.

3. "MITCHELL VIEW CALC" Disk
V-C Expanded and converted for DD, with Spreadsheet Printer.

4. "DBase-1" Disk
Disk Data Base Management. For Records of all kinds. Unlimited size and number of Records. Versatile Print Out.

5. "1987 Update Collection"
"Mail Merge", "J-Utilities", Extra Memory Utilities.

6. "Update 1989 Collection"
Utilities presented in the 1989 Update series.

SINCLAIR QL ISSUE DISKS (MDV)

1. "CABLE A-1" Disk
The "HELPER" Disk and MDV TOOL Kit. Completely manages all Media. COPY, FORMAT, Read Files Print Disk Files, A Tool Kit for for both Expanded and Stock QLs

2. "CABLE A" Disk
Two Archive Expander Programs "REFIELD" and "SCAN"

3. "CABLE B" Disk
"DBeasy", an outstanding DB Software that employs Archive.

4. "HARTUNG COLLECTION" Disk
Several Excellent Programs and Utilities. "Address File" "QSO FILE", Disk Utilities, and others.

5. "CABLE C" Disk
The "FILER" program.
Plus added selections.

6. COMING--
Programs by Al Feng
Programs by David McCall

BACK ISSUES of Update Magazine are available by year series (1987, 1988, 1989) for $18.00 while supply lasts.

EMSoft
Software for the QL that really works

We sell real world software that does real work on a QL computer, much of which is tax deductible. There are business, financial and technical applications. Most operate under one of the PSION programs.

Use any of these programs even if you're scared to death of ARCHIVE* or ABACUS*. Instructions teach what's really needed but otherwise they are transparent. All are fully editable for special requirements.

Prices are from $9.95 for QLUTter, a resident boot utility, to $2850.00 for a site license for CAM MASTER, a friendly CAD program for circular cams that generates the CNC code to a vertical mill controller. (Get the demo first; it's only $9.95 and really shows the power of the QL.

Expanded and improved versions of TRUST_FUND 1.3, MAILBAG 2.0 and QLAND_LORD 1.3 are available for $4.95 with the original medium. There's DBTutor, a flash-card program, and DBProgs with 5 useful Archive utilities. We have The_Transfer_V3, for HP compatible laser printing of DTP files and a range of other useful, hard-working software for the QL.

Order TAX-I-QL/89 by November 22 to get a $19.95 early-bird discount.

PSION's PC-FOUR for MS-DOS computers is only $99.95.
(Ever see Easel output to an HP Plotter?)

We create much of our own software but we market software by other programmers. Please call/write for details about our arrangements.

Send for our catalogue which explains things in detail.

EMSoft, P.O. Box 8763, Boston MA 02114 (617) 889-0830

Tim Stoddard, 85-48 66th Road, New York, NY 11374-5212, writes: Just received a letter from Pete Fischer about SNUG. Seems that Paul Holmgren is now President. Paul and Pete attempting to pick up the pieces, along with myself as the QL and CPM Librarian. I will supply QL and Aerco FD-68 Public Domain software at the cost of medium plus postage. Contact me directly until SNUG has the Library organized.

There is a National Sinclair ECHO on Fido Net. Access via The Serial Port BBS (313 286 0145). Lastly, after reading QL Perks & Quirks I am writing a series of articles on QJUMPS Toolkit II. You'll be suprised at what can be done- ready in a few weeks. Tim...

Thanks Tim, will be looking forward to QJUMPS. Update always willing to help SNUG. Need continual short news releases about whats goin on. -BJ_

PROMISE LAND ELECTRONICS
Dan Elliott, *Owner*
COMPUTERIZED
Servicing [redacted]
electronic equipment

Route 1, Box 117
Cabool, MO 65689
(417) 469-4571
WEEKENDS

★ OUR QL AND TS-2068 REPAIR SOURCE!
★ Highly Recommended By UPDATE - BJ.

Is the *OKIMATE 20* the *ULTIMATE* TS2068 printer?

USE W/AERCO PRINTER I/F? Yes, if an IBM parallel Plug 'n Print cartridge is used.

OKI 20 FEATURES? The OKI 20 can print in NLQ, draft, pica, elite, fine, double width, italics, sub/superscript, & underline. Plain or thermal paper (tractor/friction feed) or acetate transparencies can be used.

COLOR SCREEN$ HARD-COPY? Yes!! "COLOR COPY" supports *FULL COLOR* and b/w SCREEN$ dumps & "PAINT & COPY" features *HI-RES MODE* coloring and copying.

CUSTOM CHARACTER SETS? Yes, downline loadable character sets of custom characters and graphics can be designed and transfered to an OKI 20. The utility software "OKI D.L.C.G." allows for easy design of custom char. sets and "OKI D.L.P.G." can transform a portion of a SCREEN$ picture into a downline loadable character set.

OKIMATE 20 EXPENSIVE? No, the OKI 20 & Plug 'n Print cart. are widely available for about $180. For a print sample, info. on 2068 software for the OKI 20, and a list of OKI 20 dealers/prices, send a LSASE to:

John McMichael
1710 Palmer Dr.
Laramie, WY 82070

NEW! OKIMATE 10 SUPPORT

LKDOS SOFT-WARE

MAXCOM 300/1200 baud Terminal/BBS

This full feature modem software lets you effectivly link the modem to the disk, so you can upload or download files larger than 100K.This lets you send NMI saves and large text files as well as Basic or Code files, which other modem programs couldn't send because of memory limitations. Also files can be sent with or without a information header (Spec-term compatible) .

It also has a 64 column display and is the only 2068 modem program that will run at full 1200 baud without missing characters. Other features of the terminal mode are IBM graphic characters, auto-dialer, macro keys, 110K disk buffer for captured text, auto repeat keys.

The BBS is the most elaborate available for the 2068 and allows the remote user full access to the disk drives similar to a CPM BBS. Features of the BBS are - 300/1200 baud, operation with fast reponce even at 1200 baud, Passwords, priority levels, Use time limits, Multiple Message Bases, Uploads and Downloads are not limited by Memory, Dual Window Monitor screen, Interupt driven time out and Hangup protection, Also Keeps a User log of callers Names and Activity on disk so a printer isnt needed and its easily customised and expanded.

MAXCOM can be used with the 2050 modem or with the Z-SIO RS232 interface and a 300/1200 baud modem. (a version for the Aerco RS232 is in the works).

PRICE: $24.95

LARKEN DISK EDITOR

This program lets you examine or change any byte on a LKdos disk. It is useful in repairing damaged or glitched disks, changing the disk Name or head speed etc. You can also examine the track map and disk info headers.

The documentation for the disk editor also includes information on how to access LKdos from Machine Code and information on the structure of the disk catalog and data blocks.

PRICE: $15.00

SEQUENTIAL/RANDOM Access FILES

This utility is a ram resident extension to the LKdos operating system and lets you create, read from, or write to very large data files kept on disk. It uses the commands OPEN#, CLOSE#, PRINT#, INPUT#, and INKEY$# to access the files from BASIC. There is also a high speed search command for finding a text string within the data file. Using the TAB command, you can randomly access directly any record in a file.

The utility is easily used within your basic programs and has a lot of uses, from storing large data files, creating large spell checker dictionarys or even converting you basic programs to an ascii text file .

This utility lets you take the limits off of the data capacity of the 2068 .Files on disk can be huge !

A simple data base program is included on the disk to demonstrate the utility.

PRICE: $ 15.00

Coming Soon from Larken Electronics

** The Larken Desk Top Publisher ** Produce high quality printing (like this page) with an epson compatible printer and LKdos. The entire HI-res graphic page is kept on the disk and scrolling though the page is as easy as clicking the joystick or mouse on the icon type menu. Text and Graphic editing with multiple sized fonts will also be featured.

** Spell Checker for Tasword and M-Script ** This utility program will let you have your wordprocessor text files be checked by a 350K dictionary.

All Prices $US - Specify type of drive when ordering.

LARKEN ELECTRONICS RR#2 NAVAN, ONTARIO CANADA, K4B-1H9

(613)-835-2680

[] SPECTERM-64(TS-4.1)

300/1200 Terminal Software for T/S2068
On JLO, AERCO, Larken disk or tape.
Only $25.00 + $2.50 S&H

ALSO

[] Z-SI/O Bare Boards

RS-232c Serial Interface for T/S2068
With EBEE 2050 card
Only $24.50 + $2.50 S&H

OR BOTH FOR ONLY
$50.00 inc. S&H
(Buyers 48 customers receive, $4.50 S&H)

****

LARKEN Disk Hardware:

[ ] LARKEN Disk System

Includes LARKEN Floppy Disk Interface, LKDOS (v.3) Cartridge, Operating Manual and LKDOS Utilities' Disk.
Only $120.00 + $5.00 S&H

[ ] LKDOS Cartridge for JLO

Specially configured LKDOS (v.3) Cartridge that works with your JLO SAFE Disk hardware. Includes Operating Manual and LKDOS Utilities' Disk.
Only $65.00 + $3.50 S&H

[ ] LARKEN Ramdisk

No RAM chips are included. You can add up to 256K to this Nonvolatile Ramdisk. Easy to use as a floppy disk with LKDOS.
Only $60.00 + $3.50 S&H

[] Tandon TM101-4 Quad Disk Drives

These disk drives are brand new in factory sealed cartons. They are double sided, double density, 96 tpi (quad), full height drives. They are compatible with JLO, LKDOS, AERCO and QL disk systems.
$45.00 + $5.00 S&H

Buy 2, Get FREE Drive Cable!
Buy 3, Get 4th for Half Price!
Buy 4, Get One Disk Drive FREE!

[ ] Cable (2dr)..$12.50+$2.50 S&H

****

Hayes Compatible MODEMS

[ ] Avatex 2400 (3/12/2400)....$179
For CP/M, IBM and other systems.

[ ] Avatex 1200ho (300/1200).....$99
For T/S2068 (w/ Z-SI/O) and QL.

[ ] Avatex 1200<P> (300/1200).....$125
<P>ocket size for the Z88.

[ ] Avatex 1200<E> (300/1200).....$85
Great for SPECTERM-64!

Epson Compatible PRINTERS

[ ] Seikosha 120OAI (120/25)..$199

[ ] Seikosha 180AI (100/20)..$169

[ ] Star NX1000 (144/34)....$229

All modems add $3.50 for S&H, all printers add $10.00 for S&H (Canada double S&H on all items).

Ed Grey Enterprises
P.O. Box #2166, Inglewood, CA 90305
Info >> 213-759-7406 << Order
Call The Grey Matter BBS & RCP/M (213-971-6260)

We provide continuing support for the TS-2068 Users. Our specialties are Telecom Modems, Z-SI/O Interface, Printers, and Disk Drives for both the TS-2068 and the QL. Call for your special needs. Use our BBS for keeping updated about Sinclair Computing, Shopping, and Message traffic. Ed Grey.

The Cambridge Z88 Portable Computer

DOMINO CUBES 212 971 5638
355 WEST 39th ST.
NEW YORK CITY 10018-1401
PRESIDENT.....MIKE FINK

THE Z-88 STARTS AT $488; THEN DROPS $6 FOR EACH $100 WORTH OF RELATED ACCESSORIES YOU ORDER-STOPS AT $428

P R I C E L I S T

1.	THE Z-88 PORTABLE COMPUTER	$488
2.	32K RAM OR EPROM CARDS.	$ 45
3.	128K RAM OR EPROM CARDS.	$ 95
4.	512K RAM CARDS	$395
5.	PARALLEL PRINTER CABLE	$ 62
6.	SERIAL PRINTER CABLE (a)	$ 22
7.	PC LINK - IBM	$ 75
8.	PC LINK - MACINTOSH	$125
9.	PC LINK - QL	$ 45
10.	EPROM ERASER - 2 CARDS	$ 75
11.	EPROM ERASER - 4 CARDS	$ 95
12.	A/C ADAPTER: Z-88, WALL.	$ 15
13.	A/C ADAPTER: Z-88, CAR	$ 15
14.	MY MANUAL "A" FREE WITH Z-88	$ 10
15.	MY MANUAL "B"	$ 20
16.	SQUARE PIN PRINTER 135 CPS	$300
17.	SQUARE PIN PRINTER 200 CPS	$700
18.	PORTABLE! INK-JET PRINTER 180 CPS	$405
19.	CARRYING CASE "A"	$ 25
20.	CARRYING CASE "B"	$ 85
21.	202K PORTABLE DISK DRIVE +	$420
22.	20 HR. AA BATTERIES (4)	$4.5
23.	40 HR. AA BATTERIES (4)	$ 14
24.	5 NI-CAD 'C' BATTERIES	$ 25
25.	1200 BAUD MODEM PKG	$275
26.	3.5" DISKS, 10 PACK	$ 30
27.	RIBBON-FILM C130BK PRINTER	$5.7
28.	INK-JET INK, PLAIN or INK-JET	$ 13
29.	INK-JET STAND	$ 15
30.	COLORED INK (R-G-B)	$ 15

FREE DELIVERY UNLESS YOU ORDER A PRINTER, THEN ADD $15

♠ ♠ ♥ ♦ ♠ ♠ ♥

THE ONLY DEALER IN THE WORLD WHO GIVES YOU AN *UNDERSTANDABLE* BEGINNERS INSTRUCTION MANUAL!! AND THE ONLY DEALER SELLING AN AFFORDABLE EPROM ERABER, AND THE PLACE TO BUY THE PORTABLE DISK DRIVE TO SAVE LOADS OF MON$Y, FREE LOTTO PROGRAM WITH Z-88 PURCHASE, ALSO THE ONLY PERSON ON THE PLANET WHO CAN SHOW YOU HOW TO COLLATE, AND PRINT MULTIPLE COPIES OF MULTIPLE FILES EASILY!!! *WHILE YOU ARE ASLEEP OR AT THE MOVIES//*

♠ ♠ ♥ ♦ ♠ ♠ ♥

ALSO, THE ONLY PERSON OFFERING PROGRAMS ON CHEAP DISKS!! THEY WILL COST BETWEEN $10 AND $40

♠ ♠ ♥ ♦ ♠ ♠ ♥

THE ONLY PLACE TO GET CAR AC ADAPTERS!

♠ ♠ ♥ ♦ ♠ ♠ ♥